Hybrid Attention and Relative Average Discriminator Based Generative Adversarial Network for MR Image Reconstruction

Lencho Desalegn Yadesa

A Thesis Submitted to the Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

June 2023

Adama, Ethiopia

Hybrid Attention and Relative Average Discriminator Based Generative Adversarial Network for MR Image Reconstruction

Lencho Desalegn Yadesa

Advisor: Worku Jifara (PhD)

A Thesis Submitted to the Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

June 2023

Adama, Ethiopia

Declaration

I hereby declare that this Master Thesis entitled "**Hybrid Attention and Relative Average Discriminator based Generative Adversarial Network for MR Image Reconstruction**" is my original work. That is, it has not been submitted for the award of any academic degree, diploma, or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation.

Lencho Desalegn Yadesa _____ _____

Name of student Signature Date

Recommendation of Advisors

I, the advisor of this thesis, hereby certify that I have read the revised version of the thesis entitled "**Hybrid Attention and Relative Average Discriminator based Generative Adversarial Network for MR Image Reconstruction**" prepared under my guidance by **Lencho Desalegn Yadesa** submitted in partial fulfillment of the requirements for the degree of Masters of Science in Computer Science and Engineering. Therefore, I recommend the submission of a revised version of the thesis to the department following the applicable procedures.

Worku Jifara (PhD)	_____	_____
Major Advisor	Signature	Date

Approval Page

I, the advisor of the thesis entitled "**Hybrid Attention and Relative Average Discriminator based Generative Adversarial Network for MR Image Reconstruction**" developed by **Lencho Desalegn Yadesa**, hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Worku Jifara (PhD)	_____	_____
Major Advisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the thesis by **Lencho Desalegn Yadesa** have read and evaluated the thesis entitled "**Hybrid Attention and Relative Average Discriminator based Generative Adversarial Network for MR Image Reconstruction**" and examined the candidate during open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in Computer Science and Engineering.

_____	_____	_____
Chairperson	Signature	Date
_____	_____	_____
Internal Examiner	Signature	Date
_____	_____	_____
External Examiner	Signature	Date

Finally, approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

_____	_____	_____
Department Head	Signature	Date
_____	_____	_____
School Dean	Signature	Date
_____	_____	_____
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to the Almighty God for granting me the strength, wisdom, determination, and persistence to undertake this research.

I am sincerely grateful to my advisor, Worku Jifara (PhD), for his priceless guidance, support, and mentorship throughout this study. His expertise and insightful feedback have greatly contributed to the successful completion of this research.

I would like to extend my heartfelt appreciation to my family for their unwavering love, encouragement, and belief in my abilities. Their continuous support and understanding have been instrumental in my academic journey. I am also thankful to my friends for their encouragement, motivation, and assistance throughout the research process. Their presence has provided a source of inspiration and has made this journey more enjoyable.

Lastly, I would like to express my gratitude to all the individuals, institutions, and organizations that have contributed to this research in any way, directly or indirectly. Your support has been instrumental in the successful completion of this study.

I would like to say thanks, to whole the negative and positive people who supported me whenever in my life.

Table of Contents

List of Tables

List of Figures

List of Sample Code Snippets

List of Abbreviations and Acronyms

ADMM	Alternating Direction Method of Multipliers
CBAM	Convolutional block attention module
CC	Calgary Campinas
CS	Compressed Sensing
CNN	Convolutional Neural Network
DA	De-aliasing
DC	Data Consistency
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
HARA	Hybrid Attention and Relative Average
HARUNet	Hybrid Attention and residual U-net
MRI	Magnetic Resonance Imaging
MSE	Mean Squared Error
NMSE	Normalized Mean Square Error
PACS	Picture Archiving and Communication System
PSNR	Peak Signal-to-Noise Ratio
PI	Parallel imaging
RCA	Residual Channel-wise Attention
RAD	Relative average discriminator
RSCA	Residual Spatial and Channel-wise Attention
RAM	Random Access Memory
ReLU	Rectified Linear Unit
ResNet	Residual Network

SARA	Self Attention and Relative Average
SCA	Spatial Channel wise Attention
SR	Sampling Rate
SSA	Spatial Self Attention
SSIM	Structural Similarity Index Measure
TV	Total Variation
WGAN	Wasserstein Generative Adversarial Network
ZF	Zero filling

Abstract

Reconstructing high-quality and consistent Magnetic Resonance Images form low under-sampled k- space data were a common challenging in medical image analysis. It needs an advanced reconstruction technique to overcome those challenges. Currently, deep learning techniques outperform traditional methods such as parallel imaging and Compressed Sensing for MR image reconstruction. The DAGAN, RefineGAN, RCA-GAN and RSCA-GAN worked well in high under-sampled data, and rarely show reasonable performance when dealing with low under-sampled k-space data. In this study, we proposed a GAN-based architecture called HARA-GAN by integrating residual U-net with Hybrid attention and Relative average discriminator, to modify MRI reconstruction and reduce the noise caused by low under-sampling rates. In the generator module, the proposed method used two residual U-net with a hybrid attention mechanism. In the encoder parts of the residual U-net, it used spatial and self-attention, and in decoder side also used spatial and channel-wise attention. The proposed HARA-GAN was compared to the zero filling and RSCA-GAN on brain and knee MRI datasets using both cartesian and non-cartesian under-sampling masks. It used 12.5% sampling rates for cartesian, and 10% sampling rates for both radial and spiral under-sampling masks. The reconstructed image quality and consistency were evaluated using the peak signal-to-noise ratio (PSNR), structural similarity index measures (SSIM), and normalized mean square error (NMSE). The results indicated that HARA-GAN outperforms states of arts MR image reconstruction methods based on error maps and quantitative evaluation metrics in terms of both image quality and consistency. It improved the PSNR values for brain from 30.49 to 32.60, and for knee 32.08 to 35.89 for cartesian low under sampling rates. Thus, HARA-GAN has important implications for clinical applications, as high-quality and consistent results are crucial for accurate diagnosis and treatments of brain and knee MR images.

Keywords: HARA-GAN, Magnetic resonance imaging, Generative adversarial network, Hybrid attention, Relative average discriminator, Image reconstruction.

CHAPTER ONE

1. INTRODUCTION

1.1. Background of the Study

Magnetic Resonance Imaging (MRI) is a process of scanning tissue in body parts for medical diagnosis. During the scanning, it particularly provides images of internal organs without placing any instrument inside the body. It is also a medical imaging method with the features of non-radiation, high contrast and not requiring surgery at scanning time (Ye, 2019). However, the acquisition speeds of MRI data are affected by physical aspects (gradient strength and magnetic field strength), and physiological aspects such as nerve stimulation(Lustig et al., 2008). Since the acquisition time of MRI was long (Eo et al., 2018), the prolonged scan makes the patient feel uncomfortable and will cause motion artifacts.

The key reason for the long acquisition time is that the data is collected line-by-line in *k*-space data (frequency domain and Fourier space of the image)(Lustig et al., 2008). To overcome the prolonged scan, we had to use cartesian or non-cartesian under-sampling methods for scanning MR images. Under-sampling refers to the process of acquiring a smaller number of data points than usual, based on the desired spatial resolution and points of view (Kojima et al., 2018). It helps to make scanning faster, but it requires an advanced reconstruction technique to compensate for the missing data. Otherwise, the resulting images may show artifacts, such as aliasing, blurring, and changes in image contrast. One of the fundamental challenges in the areas of image processing and computer vision is image reconstruction, where the primary goal is to reconstruct the original image by removing noise from a noise-contaminated form of the image (J. Huang et al., 2022).

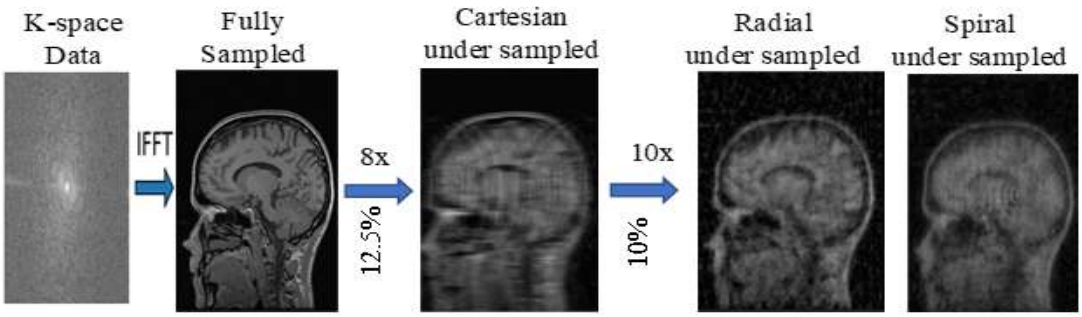


Figure 1.1: fully sampled and different under-sampling factors in MRI

Currently, Deep learning-based approaches outperform traditional methods such as parallel imaging(PI) and compressed sensing (CS) in MRI reconstruction (Esteva et al., 2021). However, they rarely show reasonable performance when dealing with under-sampled k-space data (G. Li et al., 2021). We considered an MRI reconstruction problem with k-space data input at very low under-sampled rates. To reconstruct it, we were motivated to use deep learning techniques for better reconstruction of the edges and texture of the MR images.

Compared to other techniques, Generative Adversarial Network (GAN) (Creswell et al., 2018) offers a better reconstruction technique closer to the original image details. GANs introduced an advanced learning method that balances the roles of the generator and discriminator. The generator is responsible for generating samples, while the discriminator evaluates their reality. This balance ensures that the generator does not produce overly corrupted or damaged samples while the discriminator is still able to accurately evaluate their reality.

Moreover, many studies(Quan et al., 2018; Woo et al., 2018; G. Yang et al., 2018; Q. Huang et al., 2019; Yuan et al., 2020; G. Li et al., 2021; Y. Li et al., 2021;) used attention mechanism on the areas of image reconstruction. The attention mechanism helps to extract the details of the image and allocates different weights to the attention feature maps. The Hybrid attention mechanism integrates various attention mechanisms into one framework(X. Yang, 2020), which can bring better performance for image reconstruction techniques. Convolutional block attention module(CBAM) is the first method that combined the mechanism of channel attention and spatial attention (Woo et al., 2018), making the network know not only “look what” but also “look where” by just adding a few more parameters.

The relativistic discriminator(Jolicoeur-Martineau, 2019) fills a key gap in traditional GANs by introducing a way to evaluate the reality of a generated sample relative to a reference sample. It provides the missing property that we want in GANs such as the potential to improve stability and address other issues such as mode collapse. It measures “the probability of the input data that it more realistic than a randomly sampled data of the opposing type (fake if the input is real or real if the input is fake)”, rather than measuring “the probability that the input data is real or fake”. It focused on the average of the relativistic discriminator over random samples of data of the opposing type to make the relativistic discriminator act more globally.

For MR image reconstruction, SARA-GAN (Yuan et al., 2020) introduced the self-attention mechanism and the relative average discriminator-based GAN. And also, RSCA-GAN (G. Li et al., 2021) introduced residual U-net with spatial and channel-wise attention mechanisms for better reconstruction of edges and details of an image. Based on the above studies, we proposed a GAN-based framework called HARA-GAN, which combines residual U-net with hybrid attention and relative average discriminator to modify MRI reconstruction and reduce the noise caused by low under-sampled MR images. The hybrid attention mechanism is used to improve the generator's performance. In discriminator parts, we used a relative average discriminator to improve the discriminator performance. And also we used WGAN loss(Arjovsky et al., 2017) with cyclic data consistency loss(Quan et al., 2018) to make the reconstructed MRI have high quality and consistency compared with fully sampled data.

1.2. Motivation of the Study

The main motive for this study was that the medical sector is a high-stakes application area (Yin et al., 2017). Currently, computer vision goes very crucial in the field of medical image analysis (Esteva et al., 2021). The primary objective is to enhance the patients' outcomes. To achieve this goal, deep learning can help to accelerate and simplify the analysis of medical images generated by partly or totally automating steps such as diagnosis, outcome prediction, and image reconstruction. It was also able to relieve the burden on doctors and the healthcare system. However, even though progress has been made in MR image reconstruction in research settings, surprisingly it needs improvement since it is related to the lives of humans. Because of that, we are motivated to do research on MR image reconstruction by improving the robustness of deep learning models that helps to bridge the gap between clinical practice and deep learning to use its potential for healthcare and to improve patient outcomes.

Generally, our motivation for researching MRI reconstruction started from the proposes to accelerate scan time, improve image quality, enable new imaging reconstruction, boost clinical system, and contribute to the improvement of medical imaging research. Finally, our goal is to enhance patient outcomes and contribute to the development of more effective and efficient MRI reconstruction techniques.

1.3. Statements of the Problem

Magnetic Resonance Imaging (MRI) is a widely used diagnostic tool in medical imaging. However, the lengthy scan time poses challenges for both patients and physicians. To overcome this limitation, under-sampling techniques have been employed to accelerate the imaging process. Under-sampling has advantages practically for patients and physicians in terms of reducing the MRI scanning time, costs, and as well as a patient burden (G. Li et al., 2021). However, the process of under-sampling posed challenges as it resulted in the introduction of artifacts, that led to a compromise in the quality and consistency of the reconstructed images.

The main problem addressed in this thesis paper is the inadequate performance of existing reconstruction techniques in low under-sampling rates for MRI reconstruction. Generating high-quality and consistent MR images from the under-sampled input MR image was an active and challenging research problem. Previous research has shown promising results in high under-sampling rates. However, the effectiveness of those techniques (Chen et al., 2018; Y. Li et al., 2021; Quan et al., 2018; G. Yang et al., 2018)(J. Huang et al., 2022) in low under-sampling rates remains limited based on the quantitative evaluation metrics. RSCA-GAN (G. Li et al., 2021) used residual U-net with spatial and channel-wise attention for MR image reconstruction. But it only works well at the high under-sampling rate for MR image reconstruction, and still, it was unable to achieve the best performance, especially at a low under-sampled rate based on the error maps of the reconstructed image and all quantitative metrics such as peak signal-to-noise ratio (PSNR), structural similarity index measures (SSIM), and normalized mean square error (NMSE). Additionally, it shows inadequate performance since it failed to show better image quality and removes the aliasing artifacts at the same time. And also, the previous works ignore the prior information of the discriminator's input data that improve the missing concepts from Standard GAN that leads to instability problem and mode collapse, which degrades the performance of GAN to generate high-quality images.

Generally, recent works related to MR image reconstruction have gaps to reconstruct uncorrupted or denoised scanning images from low under-sampled rate (for cartesian 12.5% and 10% for non-cartesian) k-space data to show the high image quality. The reconstructed images have noisy and details pixels are lost as shown on the error maps of the works for those low under-sampled MRI data. This thesis aims to address the reconstruction problems

associated with low under-sampling rates in MR imaging by offering novel solutions to improve image quality and consistency that enables faster scans.

1.4. Research Questions

In this study, an attempt was made to answer the following research questions.

RQ1: How hybrid attention mechanism and relative Average discriminator is implemented with GAN to improve MR image reconstruction?

RQ2: What are the effects of the hybrid Attention mechanism for reconstructing low under-sampled MR images?

RQ3: What are the effects of relative average discriminator in standard GAN for MR image reconstruction?

RQ4: What effect would Wasserstein GAN loss with cyclic data consistency loss have on the quality and consistency of MR image reconstruction?

1.5. Objective of the Study

1.5.1. General Objective

The general objective of this study is to design and implement hybrid attention and relative average discriminator-based GAN for MR image reconstruction.

1.5.2. Specific Objectives

The following specific objectives were addressed to achieve the general objective.

- ✓ To identify the method that can reconstruct low under-sampled MR images.
- ✓ To design the generator and discriminator network for the construction of edges and details of the MR image reconstruction.
- ✓ To develop the proposed architecture by improving loss functions, attention mechanism, and relative average discriminator.
- ✓ To train the model with a training dataset by enhancing the reconstruction of edges and details of the MRI.
- ✓ To evaluate the performance of the model with a test dataset using different metrics.

1.6. Significance of the Study

MR image reconstruction plays a critical role in the clinical use of medical imaging. The use of Hybrid attention and relative average-based GAN for magnetic resonance image

reconstruction has a great supporting effect on doctors to make an accurate medical diagnosis and for the patient to easily get their results. The MRI raw data is not acquired in image space, and the image reconstruction process transforms the acquired raw data into clinically interpretable images. Medical imaging has become a universal feature of clinical repetition. Modern MRI scanners are packed with advanced imaging techniques, such as parallel imaging, view-sharing methods, and spatiotemporal acceleration, as well as emerging approaches like compressed (or compressive) sensing. Because of the increasing complexity, it can be challenging to understand and apply these techniques robustly in the clinic

This work helps to reduce the complexity of constructed MR images by improving the methods of reconstruction through the deep learning algorithm. Since it tries to omit the complexity with the basic understanding of image processing, it is important for both doctors and patients. Image reconstruction is critical because flaws in the reconstruction step can compromise image quality or even the ability to generate an image. Generally, hybrid attention and relative average discriminator-based generative adversarial network for MR image reconstruction would be significant for physicians and patients to accelerate and standardize the clinical diagnosis by improving the MR image quality and consistency.

1.7. Scope and Limitations

1.7.1. Scope of the Study

The main focus of this thesis work is to reconstruct high-quality MR Images from a given k-space under-sampled compressed sensing input data.

So, this study gives the following specific controls during the reconstruction of an image:

- ✓ Brain and Knee MR image reconstruction.
- ✓ Robustness to low under-sampled rate.
- ✓ MR Image reconstruction quality and consistency improvement with GAN.

1.7.2. Limitation of the Study

This thesis work didn’t consider a high under-sampling rate for both cartesian and non-cartesian scans, detail about the diagnosis of the disease, and captioning for the MR images due to time and resource limitations

1.8. Contribution of the Study

Our main contribution involved implementing several key improvements in both the generator and discriminator parts of the GAN in the development of HARA-GAN.

- ✓ We integrated spatial and self-attention mechanisms into the encoder part and added spatial and channel-wise attention in the decoder part of the residual U-net architecture to enhance the generator's architecture.
- ✓ We implemented a relative average discriminator approach in the discriminator module to improve the discriminator performance.
- ✓ We introduced a cyclic data consistency loss mechanism alongside WGAN loss.

The above modifications enabled HARA-GAN to outperform previous methods in generating high-quality and consistent MR images measured by PSNR, SSIM, and NMSE.

1.9. Organization of the Study

This study is organized into seven chapters and they are roughly described as follows:

Chapter One: This chapter introduces the overview of the study and justifications for how and what is going to be done.

Chapter Two: In this chapter, we discussed the literature review, the traditional and deep learning-based MRI reconstruction, image-to-image translation, and related works on magnetic resonance image reconstruction with generative adversarial networks.

Chapter Three: This chapter presents a clear description of how the research methodology is done, including the dataset collection mechanisms to the model evaluation metrics.

Chapter Four: This chapter describes the proposed model in detail. The proposed architecture, loss functions, and training pseudocode have been discussed in this section.

Chapter Five: This chapter explains the implementation of the proposed solution. It also provides the environment setup for the study and the code snippet of the proposed solution.

Chapter Six: This chapter explains the results obtained from the proposed solution and discussion of the results with other work.

Chapter Seven: This chapter summarizes and concludes the work along with the result and it explores the possible future works of the study.

CHAPTER TWO

2. LITERATURE REVIEW AND RELATED WORKS

2.1. Overview of MR Image Reconstruction

Magnetic Resonance Imaging (MRI) is a medical imaging method that scans tissue in body parts without inserting any tools. MR image reconstruction is the process of reconstructing an image very closer to the original image by removing noise from a noise-contaminated form of the image. The information obtained from medical imaging is regularly reviewed to help make decisions about upcoming incidents for treatment. However, the existing approach to obtaining and reconstructing these images does not include this process. Accordingly, there is a need for improved approaches to contribute data and increase the technique of image acquisition and reconstruction.

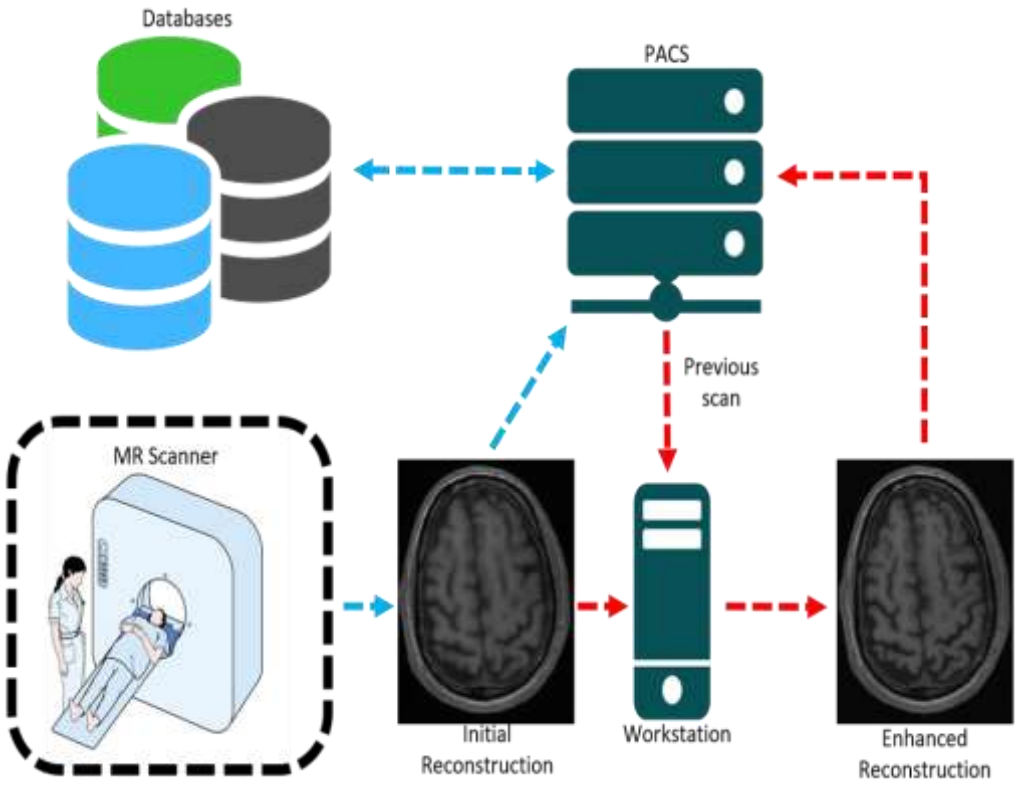


Figure 2.1: MR image acquisition and reconstruction process.

Adapted from(Souza et al., 2018)

The route indicated by the red arrows will only be activated if there is a prior MRI scan available. PACS, which stands for Picture Archiving and Communication System, is involved in this process.

2.1.1. MR Image Acquisition and K-space Filling

The process of obtaining MR images involves filling up a data space referred to as k-space, which can be filled up in multiple ways before being converted into an image via an inverse Fourier transform(Lazarus et al., 2019). The way k-space is filled up including its pattern and sequence is known as the k-space trajectory. There are two different approaches which are Cartesian and non-cartesian (radial and spiral) ways for filling up k-space as depicted in **figure 2.2**, with each straight or curved line representing a separate readout. To make the visualization clearer, the readouts are shown in solid and dashed lines alternatively. Cartesian scanning which acquires data as rectilinear lines through k-space is the most common technique used clinically today.

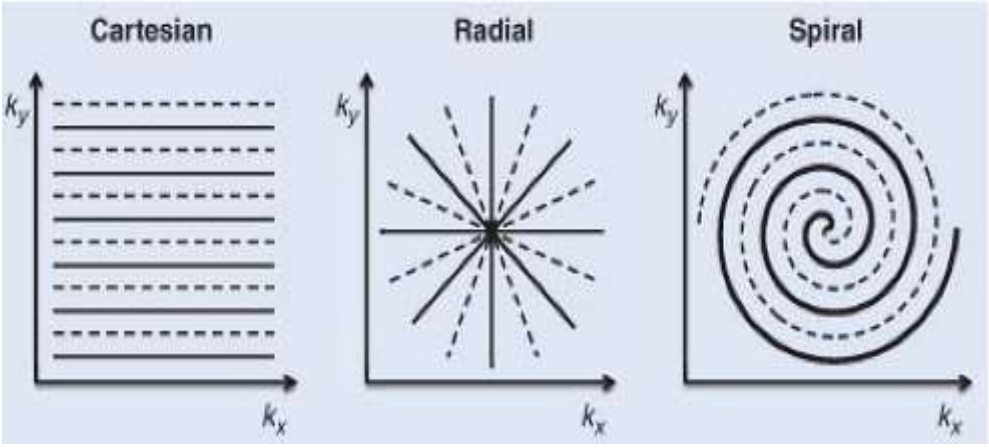


Figure 2.2: Methods of filling K-Space MRI data

Cartesian k-space filling(Paschal & Morris, 2004) is the most straightforward and commonly used trajectory. It samples k-space along Cartesian grid lines, acquiring data in a rectilinear pattern. This trajectory is simple to implement but may suffer from artifacts related to motion and aliasing.

The mathematical expression for 2D Cartesian k-space filling pattern can be represented as follows:

$$k_x = \left(-\frac{N_x}{2}, -\frac{N_x}{2} + 1, \ldots, \frac{N_x}{2} - 1\right) \quad (2.1)$$

$$k_y = \left(-\frac{N_y}{2}, -\frac{N_y}{2} + 1, \ldots, \frac{N_y}{2} - 1\right) \quad (2.1)$$

In addition to Cartesian scanning (left), there are two non-cartesian ways to fill up k-space(Paschal & Morris, 2004), such as using radial (middle) or spirals (right) as shown in

Figure 2.2. Non-Cartesian MR imaging, on the other hand, often employs radial lines or spiral patterns for filling up k-space. Currently, there is growing interest in exploring non-cartesian k-space trajectories, which are beginning to gain acceptance or an emerging adoption in clinical practice.

Radial k-space filling samples k-space along radial lines emanating from the center of k-space. This trajectory is efficient for high-speed imaging and is more robust against motion artifacts. It is particularly useful in dynamic imaging and real-time applications.

$$k_x = R * cos(\theta) \quad (2.3)$$

$$k_y = R * sin(\theta) \quad (2.4)$$

Here, R represents the radial distance from the k-space origin, and θ represents the angle of the radial line.

Spiral k-space filling follows a spiral trajectory, starting from the center and expanding outward. This trajectory provides higher sampling efficiency, allowing faster acquisitions and reduced scan times. Spiral trajectories are commonly used in functional MRI (fMRI) studies.

$$k_x = R * cos(\alpha * \theta) \quad (2.5)$$

$$k_y = R * sin(\alpha * \theta) \quad (2.6)$$

Here, R represents the radial distance from the k-space origin, and α controls the spacing between successive turns of the spiral.

The overall filling k-space trajectories for both cartesian and non cartesian are shown as following:

$$S(kx, ky) = \sum_{i=1}^{n} (kx - kxi, \quad ky - kyi) \quad (2.7)$$

2.2. Traditional MR Image Reconstruction Techniques

2.2.1. Parallel Imaging

Parallel imaging is a technique used in magnetic resonance imaging (MRI) to accelerate the acquisition of image data(Hamilton et al., 2017). It is particularly useful in reducing scan time, improving spatial resolution, and mitigating motion artifacts. In a conventional MRI scan, the data acquisition process involves measuring the magnetic resonance signal from

the entire region of interest using multiple receiver coils. However, this can be time-consuming, especially for high-resolution images or when scanning moving organs.

Parallel imaging addresses this limitation by exploiting the spatial sensitivity information provided by multiple receiver coils. Instead of acquiring data from all the coils simultaneously, parallel imaging techniques acquire a reduced amount of data from a subset of the coils and then use mathematical algorithms to reconstruct the full image.

The key idea behind parallel imaging is that different receiver coils have varying sensitivities to the underlying anatomy(Deshmane et al., 2012). By exploiting these coil sensitivities, it is possible to estimate the missing k-space data points (the Fourier space representation of the image) and reconstruct the full image with fewer acquired data points. The most commonly used parallel imaging technique is called sensitivity encoding (SENSE). SENSE uses the knowledge of the coil sensitivities to under-sample the k-space data and then employs an iterative reconstruction algorithm to estimate the missing data and reconstruct the image. Another widely used parallel imaging method is known as generalized auto-calibrating partially parallel acquisitions (GRAPPA)(Griswold et al., 2002). GRAPPA uses a calibration scan to measure the coil sensitivities and uses this information to reconstruct the under-sampled k-space data.

Parallel imaging techniques have several advantages, including reduced scan time, improved spatial resolution, and decreased susceptibility to motion artifacts. However, there are also some limitations, such as increased noise levels in the reconstructed images and potential artifacts due to inaccurate coil sensitivity estimation.

2.2.2. Compressive Sensing MRI

Compressive sensing is a signal processing technique that allows the reconstruction of a signal or image from a significantly reduced number of measurements or samples(Lustig et al., 2008). CS has shown promising results in the field of magnetic resonance imaging by reducing acquisition time or improving image quality with fewer measurements. In conventional MRI, data acquisition is performed by uniformly sampling the entire k-space, which can be time-consuming and sensitive to motion artifacts. CS provides an alternative approach by exploiting the sparsity or compressibility of MRI images.

In compressive sensing MRI, the image is acquired using a reduced number of random or pseudo-random measurements(Jaspan et al., 2015). These measurements are typically

obtained by under-sampling the k-space, which is the Fourier domain representation of the image. The under-sampling is done in a way that preserves the key information required for image reconstruction.

Compressive sensing MRI offers several advantages, including reduced scan time, improved spatial resolution, and decreased sensitivity to motion artifacts(Ye, 2019). However, there are also challenges associated with CS-MRI, such as increased computational complexity, potential artifacts due to under-sampling, and the need for careful selection of appropriate acquisition and reconstruction parameters.

Researchers continue to explore and refine compressive sensing techniques for MRI, aiming to optimize the trade-off between scan time reduction and image quality. CS-MRI has the potential to revolutionize MRI by enabling faster scans, reducing patient discomfort, and facilitating dynamic imaging of moving organs or real-time imaging applications(Jaspan et al., 2015).

2.3. Deep Learning-based MR Image Reconstruction

In recent years, the study on MR image reconstruction is very popular with the developments of deep learning approaches. MRI has a very slowly gaining speed because data samples directly take from K-space. Slow scanning has an effect due to patient movements, such as impediment cardiac pulsation, respiratory promenade, and gastrointestinal peristalsis (Shende et al., 2019). To improve the performance of reconstructed MR images several studies were performed by using different methods and Algorithm.

Subject to data consistent restrictions, the algorithm (Lustig et al., 2005) proposed to minimize the L1/wavelet penalty norm of a modified image. Another approach is to apply a total variation (TV) penalty in place of the L1/wavelet penalty because TV penalization performs marginally better than the L1/wavelet penalty for MR image reconstruction (Lustig et al., 2008). The alternate direction method was used by RecPF FCSA (J. Huang et al., 2010)to reduce a linear mix of L1/wavelet and TV norm regularization. The quality of the reconstructed images was further enhanced by FCSA, which used an iterative framework to weight the answers to the L1/wavelet and TV norm subproblems. These works focus on developing novel optimization algorithms and objective functions but ignore the computationally inefficient and time-consuming process until convergence.

Additionally, the objective function heavily depends on data prior knowledge, which is typically lacking in real-world problems. Due to their superior feature representation

capabilities, deep learning-based MRI reconstruction techniques have lately received extensive research to address these issues (yang et al., 2016)). To improve a universal CS-based MRI model, iterative processes in the Alternating Direction Method of Multipliers (ADMM) algorithm were initially described as a method for using neural networks in MRI reconstruction (Boyd et al., 2011). DC-CNN (Schlemper et al., 2017) proposed a deep cascaded convolution neural network with a data consistency unit that improves MRI reconstruction performance.

2.4. Image-to-Image Translation

Image-to-image translation has gained a lot of attention and advancement in computer vision and image processing. The goal of image-to-image translation is to convert an input image from one representation to another, such as translating a grayscale image to a colored one, or translating a photograph of a face to a cartoon-style representation, pose estimation, image synthesis, segmentation, image reconstruction/restoration and style transfer (Pang et al., 2022). The process involves learning a mapping between the input and output images and applying that mapping to generate the desired output image. Medical image reconstruction(J. Huang et al., 2010) is often considered a form of image-to-image translation or a pixel-to-pixel translation task, as the goal is to produce an accurate and visually appealing representation of an image from a limited or corrupted input. It involves restoring missing or degraded information, reducing noise, or filling in missing pixels, among other tasks.

Recently, GAN-based techniques have gained popularity in image-to-image translation and have generated pleasing outcomes (Emami et al., 2021). With the help of paired images from the source and target domains, conditional GAN (cGAN) was utilized in pix2pix(Knyaz et al., 2019) to develop a translation from a source image to an output image. CycleGAN (Jay et al., 2017) introduced image-to-image translation challenges without paired samples. MR image reconstruction is often formulated as an image-to-image translation problem (Pang et al., 2022), where the input is the raw MR data and the output is the reconstructed MR image. The process involves learning a mapping between the input and output images and applying that mapping to generate the desired output image. When performing an image-to-image translation task, a generative model can simulate the distribution of the target domain by generating convincing "fake" data, such as translated images, that appear to be taken from the distribution of the target domain(H. M. Zhang & Dong, 2020).

Since an image-to-image translation job tries to learn the mappings between various picture domains, the generative models are directly tied to how to represent these mappings to provide the desired outcomes. The generative model (G. Li et al., 2021; Quan et al., 2018; Schlemper et al., 2017) assumes that data is created by a particular under-sampling mask that is defined by two parameters (Cartesian distribution sampled line-by-line or non-Cartesian i.e., spiral and radial), and it approximates that underlying distribution with particular deep learning algorithms.

2.5. Generative Adversarial Network

Generative Adversarial Networks (Goodfellow et al., 2014) were recently introduced as a novel way to train generative models. In their work, they introduce the conditional variant of generative adversarial networks, which can be created by simply passing the data, y, they desire to condition on to both the generator and discriminator. Generative adversarial networks were a system or architecture where two neural networks are competing with one another to generate a variation in the data being generated. In generative adversarial networks, there is a generator and discriminator module(Creswell et al., 2018). Therefore, as implied by its name, generative refers to the creation of false data, and adversarial refers to the idea of a competition between the discriminator and the generator for winning each other's, in which case the generator would strive to deceive while the discriminator would try to avoid being deceived. The sophisticated model is being learned and trained simultaneously by the two of them.

The discriminator D(x) in the GAN is a classifier that is used to distinguish between true and data that has been generated by the generator, and it can be any network design that is compatible with the classifier (model). The discriminator's output probability were ranging from [0, 1], and its inputs are the generated image and the genuine input (G. Li et al., 2021). The generator G(z) learns how to produce a fake image that could lead the discriminator to mistake the output for the actual thing and classify it as such. According to (Chen et al., 2018), the generator typically receives a random noise input (z) and attempts to construct a meaningful output from it.

Generative adversarial networks (GANs) are a new technique for semi-supervised and unsupervised learning. They achieve this through implicitly modeling high-dimensional data distributions. Proposed in 2014 (Mirza & Osindero, 2014), they can be characterized by training a pair of networks in competition with each other. Crucially, the generator has no

direct access to real images, it learns only through its interaction with the discriminator. The discriminator has access to both samples drawn and the synthetic samples from the stack of real images. The error signal to the discriminator is provided through the simple ground truth of knowing whether the image came from the real stack or the generator (Creswell et al., 2018).

Generally, GAN training happens in two stages. First, freeze the generator and train the discriminator, delaying generator training and allowing only a forward pass with no backward propagation. Second, freezing the discriminator and teaching the generator to provide results that are more plausible and would deceive the discriminator. The general steps for training generative adversarial networks, which is an iterative process, are as follows.

- ✓ Train the Discriminator:
  - Take a random mini-batch of real examples: x.
  - Take a mini-batch of random noise vectors z and generate a mini-batch of fake examples: $G(z) = x_{fake}$.
  - Compute the reconstruction losses for $D(x)$ and $D(x_{fake})$, and backpropagate the total error to update $\theta^{(D)}$ to minimize the reconstruction loss.
- ✓ Train the Generator:
  - Take a mini-batch of random noise vectors z and generate a mini-batch of fake examples: $G(z) = x_{fake}$.
  - Compute the reconstruction loss for $D(x_{fake})$, and backpropagate the loss to update $\theta^{(G)}$ to maximize the reconstruction loss.

Multilayer networks with convolutional and/or fully connected layers are commonly used to represent the generator and discriminator. The generator and discriminator networks must be differentiable, but they do not need to be directly invertible.

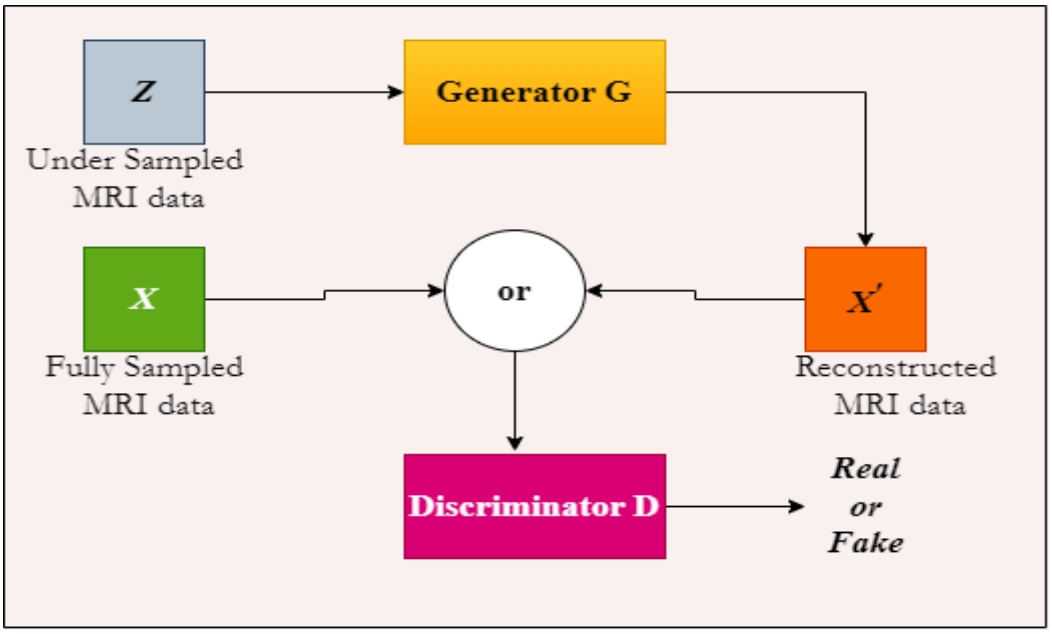


Figure 2.3: GAN Architecture for MRI reconstruction. Adapted from (Creswell et al., 2018)

2.6. Residual U-Net

Residual U-Net (Z. Zhang et al., 2018) is a variant of the U-Net architecture for image segmentation that utilizes residual connections to improve the flow of information and alleviate the vanishing gradient problem. In MR imaging, residual connections (He et al., 2016) can be added to a U-Net architecture(Ronneberger et al., 2015) to help reduce the impact of measurement noise and improve the quality of the reconstructed image. The residual connections allow the network to directly incorporate information from earlier layers into the reconstruction process, which can help to better preserve fine details and reduce artifacts in the reconstructed image. Thus, residual U-Net can be seen as a tool for improving the performance of MR image reconstruction algorithms. The residual connections allow the network to directly pass the information from earlier layers to later ones, providing an easier path for gradients to flow and making it easier for the network to learn. This can help improve the accuracy of the network and make it more effective at segmenting images. Residual U-Net can be used for magnetic resonance (MR) image reconstruction, which is the process of recovering the underlying structure of an MR image from incomplete or noisy measurement data (G. Li et al., 2021).

Residual U-Net is used for MR image reconstruction because it offers several advantages over traditional image reconstruction algorithms (Q. Huang et al., 2019). It improved accuracy by using residual connections, and Residual U-Net can help to reduce the impact

of measurement noise and improve the quality of the reconstructed image. Residual U-Net can help to preserve fine details in the reconstructed image, which is important for medical imaging applications where small structures and details can provide important diagnostic information. And also, residual U-Net is more robust to input variations, which can be particularly important in MR image reconstruction where measurement noise and other factors can cause significant variations in the input data(G. Yang et al., 2018). Residual U-Net can be trained end-to-end, which means that it can learn the mapping between the measurement data and the underlying image structure directly, without the need for intermediate steps or manual tuning of parameters. Generally, the use of residual connections in U-Net provides an improved way of integrating information across multiple layers in the network, which can lead to better performance in MR image reconstruction tasks.

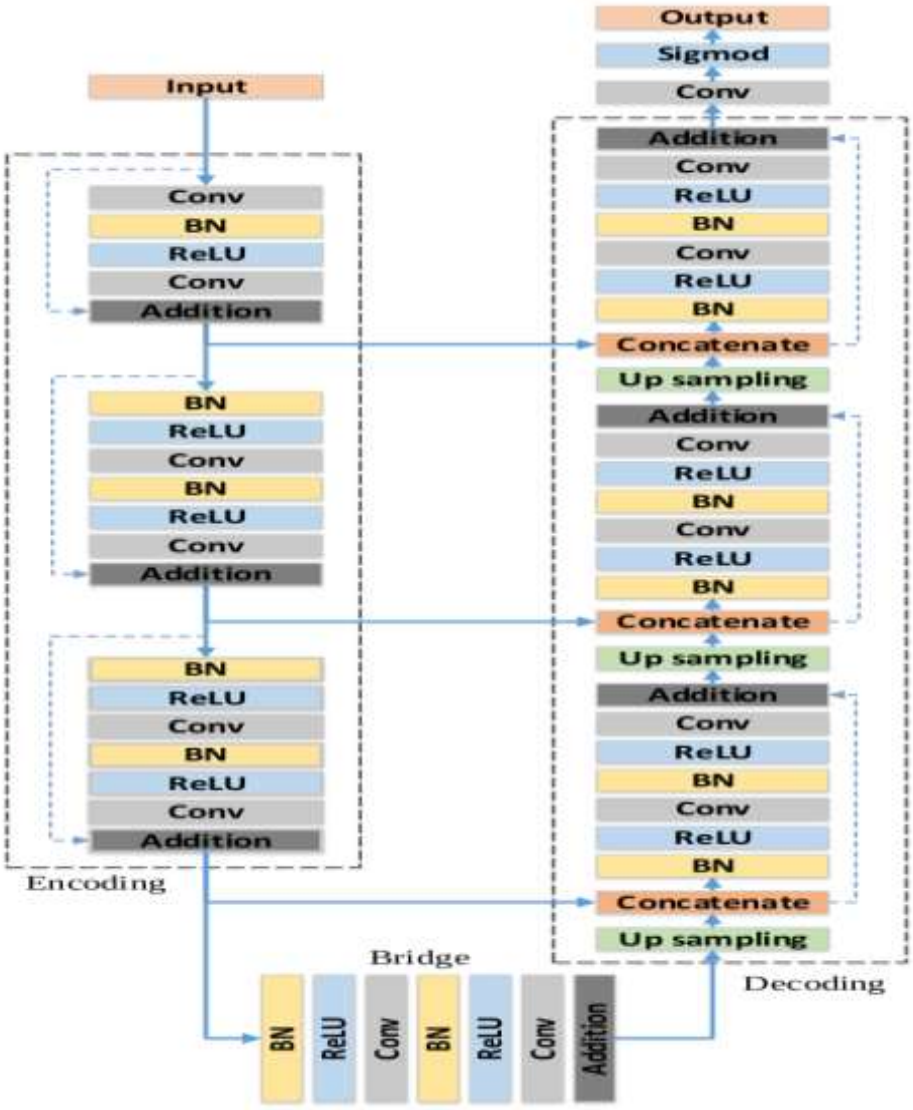


Figure 2.4: Residual U-Net Architecture. Adapted from (Z. Zhang et al., 2018)

Residual U-Net for MR image reconstruction works by using a neural network architecture that is similar to a traditional U-Net (Ronneberger et al., 2015), but with the addition of residual connections. The basic idea behind the residual U-Net architecture is to use the residual connections to provide a direct path for information to flow from earlier layers to later ones. This helps to reduce the impact of measurement noise and other factors that can cause variations in the input data.

The input to the residual U-Net is the measurement data obtained from the MR imaging system, which is then processed through a series of convolutional and up-sampling layers (G. Li et al., 2021). The residual connections allow information from earlier layers to be directly passed to later ones, which can help to reduce the impact of measurement noise and improve the quality of the reconstructed image. The output of the residual U-Net is an MR image that is reconstructed from the measurement data(G. Yang et al., 2018). The network can be trained end-to-end, which means that it can learn the mapping between the measurement data and the underlying image structure directly, without the need for intermediate steps or manual tuning of parameters (Ronneberger et al., 2015).

The structure of residual U-Net consists of two main parts i.e., the encoder and the decoder(Z. Zhang et al., 2018). The encoder is responsible for extracting features from the input measurement data and reducing the spatial resolution of the data. It typically consists of several blocks of convolutional and pooling layers, where each block reduces the spatial resolution and increases the number of feature maps. The decoder is responsible for up-sampling the feature maps from the encoder and producing the final reconstructed image. It typically consists of several blocks of up-sampling and convolutional layers, where each block increases the spatial resolution and reduces the number of feature maps.

In addition to the encoder and decoder, residual U-Net also includes residual connections that allow information from earlier layers to be directly passed to later ones (Quan et al., 2018). These residual connections can be added at different points in the network and can take different forms, depending on the specific implementation (Zeng et al., 2019). Generally, the structure of residual U-Net is designed to provide a way of integrating information across multiple layers in the network and improving the flow of information through the network. This can lead to improved performance in MR image reconstruction tasks.

2.7. Hybrid Attention Mechanism

Hybrid attention(Jiang et al., 2022) is the combination of two or more attention in one framework to improve the model performance in the areas of computer vision and image processing. Attention mechanisms are initiated from the inquiries of human visualization. In cognitive science, only a part of all visible information is perceived by human beings due to the bottlenecks of information processing. Encouraged by this visual attention mechanism, researchers have worked to develop a model of visual selective attention to mimic the human visual perception process. This allowed them to better understand how people's attention is distributed while viewing both still images and moving pictures, and also allow them to expand the scope of their research (X. Yang, 2020).

Recently, significant advances in attention mechanisms have been realized in the areas of image and natural language processing. Attention methods have been shown to enhance model performance, and they are also comparable with the perceptual process of the human brain and eyes. The majority of research mixing deep learning with visual attention processes in the area of computer vision, for instance, focuses on the application of masks. The fundamental idea behind a mask is that a new layer with new weights is used to identify the important features in the image data. Deep neural networks are capable of learning which parts of each new image require attention and can do this through training and learning.

The concepts of soft attention and hard attention were developed from this initial concept. The soft attention technique is differentiable and continuous, and it is implemented using gradient descent. Hard attention mechanisms are not differentiative, which is frequently accomplished through reinforcement learning and encouraged by the benefit function to make the model pay closer attention to the specifics of some areas.

Soft attention mechanisms(X. Yang, 2020) can be divided into four categories such as self-attention, spatial attention, channel-wise attention, and mixed/hybrid attention. The hybrid attention mechanisms used in this study were spatial and self-attention in encoder parts of the U-net and Spatial and channel-wise attention in decoder parts of the U-net.

2.7.1. Spatial and Self-attention

In neural networks, spatial attention and self-attention are both utilized as strategies to aid the model in focusing on particular input regions or previously hidden states(R, 2021). The self-attention mechanism is the process by which comparable pixels are used for training and prediction while dissimilar pixels are ignored. The term "spatial attention" also describes

a model's capacity to selectively concentrate on particular regions of an image or other two-dimensional input (X. Yang, 2020). This might be helpful for jobs where the model must recognize particular objects or regions in a picture, like object identification or image captioning. When the model needs to identify particular objects or regions in an image, such as during the process of object recognition or image reconstruction and picture captioning, this can be very helpful.

The model uses self-attention to focus on specific parts of the input sequence and weigh the importance of different parts of the input before generating output. A process employed in sequence-to-sequence models like Transformer is self-attention, also referred to as spatial-attention. Before producing output, the model uses self-attention to concentrate on specific segments of the input sequence and assess the relative relevance of the segments. It is employed to enhance the model's performance model to handle long-term dependencies in the input. The spatial self-attention module influences a self-attention mechanism proposed by (Osman AM, 2018), which accumulates local attention through a SoftMax function. We extend this idea to attend to the spatial pixel positions of the original features and apply feature aggregation to obtain spatial self-attention feature maps. The spatial and self-attention module effectively acquires discriminate slice localization and dramatically boosts performance.

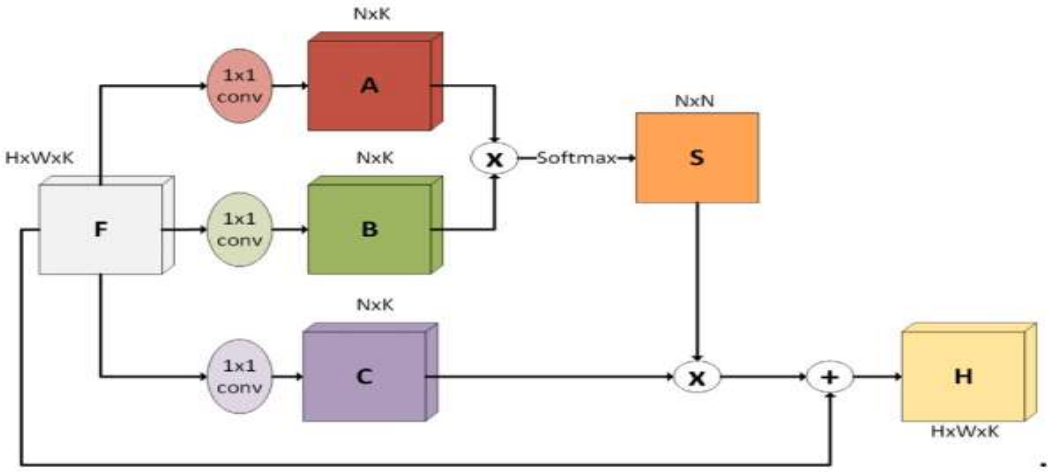


Figure 2.5: The SSA module. Adapted from ((R, 2021))

It denotes $+$ as element-wise addition and X as element-wise multiplication.

2.7.2. Spatial and Channel-wise attention

Spatial and channel-wise attention(G. Li et al., 2021) was a combination of spatial and channel-wise attention in a single framework. Different attention techniques employed in neural networks include spatial attention and channel-wise attention. The ability of a model to selectively focus on particular regions of an image or other two-dimensional input is referred to as spatial attention, as I indicated previously. This can be achieved by using a 2D convolutional filter to create a "heatmap" of attention on the image, with high values denoting regions of interest for the model.

The attention that is focused on the channels of a feature map rather than the spatial dimensions is referred to as "depth-wise attention" or "channel-wise attention."(Woo et al., 2018) This is helpful for convolutional neural networks (CNNs) because the model must focus narrowly on a few channels in a feature map to perform well. A 1D convolutional filter that is applied along the channel dimension can be used to implement channel-wise attention. Using this, it is possible to create a "channel-wise attention map" that illustrates the feature map channels that are most crucial for the work at hand.

Each image in a neural network has three channels as its initial representation (R, G, B). Each channel will produce additional channels with distinct information after being treated by various convolution kernels. If weights are given to each channel to demonstrate the relationship between each channel and important information, a larger weight indicates higher importance, and the relevant channel should receive more attention(G. Li et al., 2021).

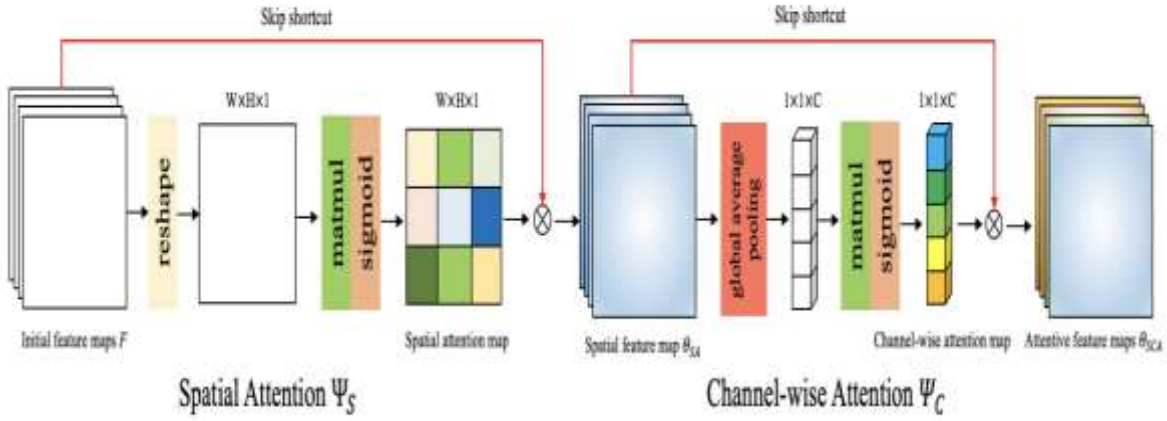


Figure 2.6: Architecture of spatial and channel-wise attention. Adapted from((G. Li et al., 2021))

2.8. Relativistic Average Discriminator

A relativistic average discriminator is an element of a Generative Adversarial Network (GAN) that is trained to distinguish between real and fake data(Jolicoeur-Martineau, 2019). The discriminator is presented with a sample of data and must determine whether it is real or generated by the GAN's generator. The discriminator must be trained to be extremely accurate to force the generator to produce more realistic images to deceive the discriminator and enable the GAN to generate realistic images.

According to (Jolicoeur-Martineau, 2019), Using the relativistic discriminator, standard GAN can be fixed and enhanced. It greatly enhances GANs' stability and data quality without generating any computational costs. The relative discriminator provides the missing property that we want in GANs (i.e., *G* influencing *D*), its interpretation is different from the standard discriminator(Yuan et al., 2020). It is now measuring "the probability that the input data is more realistic than a randomly sampled data of the opposing type (fake if the input is real or real if the input is fake)". Rather than measuring "the probability that the input data is real", To make the relativistic discriminator act more globally, as in its original definition, our initial idea focused on the relative average discriminator over random samples of data of the opposing type.

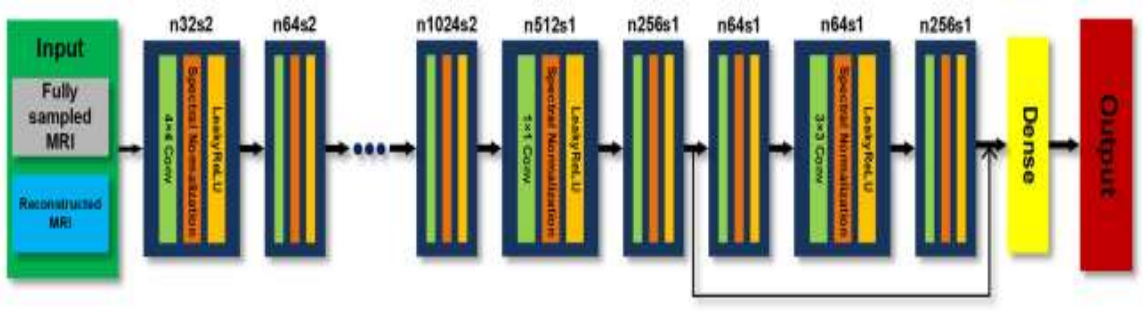


Figure 2.7: The relative average discriminator. Adapted from((Yuan et al., 2020))

We employed the relative average discriminator (Jolicoeur-Martineau, 2019), and we think the discriminator should fully utilize the previous information to determine the chance that the given full sample MRI is more realistic than the reconstruction MRI, on average. And also, explanatory studies were conducted using existing schemes and a recently introduced deep learning system.

2.9. Related Work

Recently, with the development of generative adversarial network (GAN) (Creswell et al., 2018), several methods (Y. Li et al., 2021; Mardani et al., 2019; Quan et al., 2018) are proposed to improve image quality with modified CS-MRI reconstruction. Moreover, many studies have applied attention mechanisms to the image reconstruction field. The attention mechanism extracts the details of the image and assigns different weights to the hybrid (channel, spatial, and self-attention) feature maps.

DAGAN (G. Yang et al., 2018) proposed a novel deep de-aliasing generative adversarial network (DAGAN) for fast CS-MRI reconstruction. In DAGAN the details of the brain cannot be distinguished well. DAGAN was used for MR image reconstruction by training a generator network to generate high-quality images from under-sampled MRI data. The generator network is trained using a loss function that encourages the generated images to match the ground truth images as closely as possible. Refine GAN (Quan et al., 2018) is a combination of cyclic consistency, U-net(Ronneberger et al., 2015), and deep residual CNN (X. Li et al., 2018). Refining a GAN involves making improvements to its architecture or training process to enhance its performance in generating high-quality outputs. Refine GAN removes the aliasing artifacts well and shows acceptable reconstruction results, but it can't achieve better image quality. Two residual U-nets were used as a generator to perform end-to-end MRI reconstruction, which deepens the network of the generator.

PTGAN (Chen et al., 2018) used U-Net as a generator model that can effectively preserve the structure of images for MRI reconstruction. It adds k-space frequency domain loss, frequency edge loss, and frequency center loss to the generator loss to achieve high organizational similarity and resolution after MRI reconstruction. RCA-GAN (Q. Huang et al., 2019) had to use channel-wise attention in U-net up-sampling for MR image reconstruction. In the context of U-Net, channel-wise attention allows the network to focus on important features in the feature maps at each stage of the up-sampling process. And also, other works proposed a deep cascading of the U-net structure, which reduces reconstruction error (Sun et al., 2019). SARA-GAN (Yuan et al., 2020) developed the self-attention mechanism and the relative average discriminator-based GAN for under-sampled k-space data CS-MR image reconstruction. It is a modification of the standard GAN architecture that uses a relativistic discriminator to provide more informative feedback to the generator. Rather than simply classifying images as real or fake, the SARA-GAN discriminator

compares the real and generated images in a relative sense, providing feedback to the generator on how it can improve the realism of the generated images. Using the long-range global dependence constructed by the self-attention module, this technique can reconstruct images with more realistic image details and good quantitative metrics with quality compromised in high under-sampled k space data.

RSCA-GAN (G. Li et al., 2021) used residual U-net with spatial and channel-wise which is a type of Generative Adversarial Network (GAN) that is designed for MR (Magnetic Resonance) image reconstruction. These include residual blocks, spatial and channel-wise attention mechanisms, and a cyclic data consistency loss function. Residual blocks are a type of neural network block that allows for the efficient training of deep neural networks. In RSCA-GAN, the generator network is composed of multiple residual blocks, which allow for the efficient propagation of information through the network. The performance of the reconstruction results in PSNR, SSIM, and NMSE values are 33.14, 0.91, and 1.20 respectively. However, both the error map and the quantitative value indicate that they can't achieve the best performance, especially at a low under-sampled rate.

Table 2.1: Summary of related work

Numbers.	**Methods**	**Objective**	**Contribution**	**Gaps**
1.	DAGAN (G. Yang et al., 2018)	✓ To develop a novel deep de-aliasing generative adversarial network for fast CS-MRI reconstruction. ✓ To achieve better reconstruction details.	✓ Provides U-Net architecture with skip connections for the generator network. ✓ The adversarial loss is coupled with a novel content loss defined by pre-trained deep convolutional networks from the VGG.	The details of the brain cannot be distinguished well
2.	PT-GAN (Chen et al., 2018)	✓ To generate better MR image quality using U-net architecture. ✓ To achieve high organizational similarity and resolution after picture conversion.	✓ add k-space frequency domain loss, including frequency edge loss and frequency center loss, to the generator loss.	Can't achieve better performance in constructing details of an image and removing artifact
3.	SARA-GAN (Yuan et al., 2020)	✓ To reconstruct images with more realistic image details and good quantitative metrics with quality compromised in high under-sampled k space data.	✓ Provides relative average discriminator in discriminator parts of Generative adversarial networks to improve the discriminator performance.	Challenged to reconstruct well, when the input image is low under sampled MRI.

4.	Refine GAN (Quan et al., 2018)	✓ To remove the aliasing artifacts and achieve better edge reconstruction.	✓ Provides cyclic data consistency loss ✓ Add cyclic data consistency loss in generative adversarial networks with U-net for faithful interpolation in the given under-sampled k-space data.	It can't achieve better image quality for MR image reconstruction.
5.	RCA-GAN (Q. Huang et al., 2019)	✓ To generate better image quality using channel attention in encoder parts of U-net.	✓ Provides a channel-wise attention mechanism with residual U-Net ✓ Provides dual residual U-Net	It can't remove the aliasing artifacts well
6.	RSCA-GAN (G. Li et al., 2021)	✓ To generate better image quality using Spatial channel attention in decoder parts of U-net MR. ✓ To improve the loss function in generator parts of the GAN	✓ Provides a spatial and channel-wise attention mechanism with residual U-Net for MR image reconstruction. ✓ Provides the new total loss function by mixing the generative adversarial loss and cyclic data consistency loss	It fails to show better image quality and removes the aliasing artifacts as required, in low under-sampling rates.

2.9.1 Summary of Related Work

Many recent works (Ahishakiye et al., 2021) used the availability of K-space data to solve the issues related to MR image reconstruction for different MR images to train the model using generative adversarial networks. The Refine GAN, RCA-GAN, SARA-GAN, and RSCA-GAN can reconstruct brain images well. RCA-GAN shows better image quality than RefineGAN since it adopts channel attention. But it can't remove the aliasing artifacts well for brain MRI reconstruction. The SARA GAN techniques are not working well when the inputs of the networks are low under sampled MRI. The RSCA-GAN incorporates some advanced techniques to improve MR image reconstruction performance. It works well in MR image reconstruction, but still can't achieve the best performance, especially in low under-sampled rates of MRI data. And also, it is under compromise since it fails to show better image quality and removes the aliasing artifacts at the same time.

Based on the above studies, we proposed HARA-GAN that combines the Hybrid-Attention mechanism with a Relative Average discriminator for reconstructing under-sampled MR images. HARA-GAN were able to reconstruct MR image with edge and details of images by showing better image quality and removing the corrupted artifacts well.

2.9.2. Baseline Work

We selected "A Modified Generative Adversarial Network Using Spatial and Channel-Wise Attention for CS-MRI Reconstruction" (G. Li et al., 2021) as baselines of our works based on the model performance and generated outcomes when compared to others.

RSCA-GAN (Residual Spatial Channel Attention Generative Adversarial Network) (G. Li et al., 2021) is a type of Generative Adversarial Network (GAN) that is designed for MR (Magnetic Resonance) image reconstruction. It used two residual U-net with spatial and channel-wise attention mechanisms at generator parts of the network and standard discriminator in discriminator sides of the generative adversarial network.

Spatial and channel-wise attention mechanisms are used to help the network focus on important regions of the input data when generating the output. These mechanisms allow the network to selectively attend to different parts of the input data, improving the quality of the generated images. The perceptual loss function is a modification of the standard GAN loss function that has been shown to lead to more stable and effective training of GANs. In RSCA-GAN, the perceptual loss function is used to encourage the generator to produce images that are indistinguishable from the ground truth images.

The findings demonstrate that RSCA-GAN is capable of achieving the best possible reconstruction outcomes, regardless of whether the sampling mask is Cartesian or non-Cartesian. This suggests that the reconstruction procedure is comparable to the denoising of images and is not influenced by the specific contents of the image.

In a series of experiments, the authors demonstrated that RSCA-GAN outperformed other state-of-the-art methods for MR image reconstruction in terms of both visual quality and quantitative metrics for different under-sampling rates. Generally, the combination of residual blocks, attention mechanisms, and perceptual loss function makes RSCA-GAN a powerful tool for MR image reconstruction.

The residual spatial and channel-wise attention-based generative adversarial network outperformed other reconstruction methods for brain and knee test images, as evaluated by quantitative metrics such as Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Normalized Mean Squared Error (NMSE). This superiority was consistent across various sampling rates, including 10%, 20%, and 30% for non-cartesian sampling, and 12.5%, 16.7%, and 25% for cartesian sampling.

CHAPTER THREE

3. RESEARCH METHODOLOGY

3.1. Chapter Overview

This section describes data collection techniques, software, and hardware tools used in the successful accomplishment of the research. So, to achieve the objective of this study the following methods and techniques were employed.

The methodology employed in the implementation of hybrid attention and relative average discriminator based on GAN for MR image reconstruction as well as the mathematical formulation of the evaluation metrics are described in this chapter. This chapter goes through the methodology, dataset, and experimental measures used to address the study questions in further detail.

The following figure 3.1 shows how the MR image dataset is gathered for the purpose of implementation. The diagram consists of different stages from collecting, preprocessing, and utilizing the datasets for training and testing the model and finally evaluating the model.

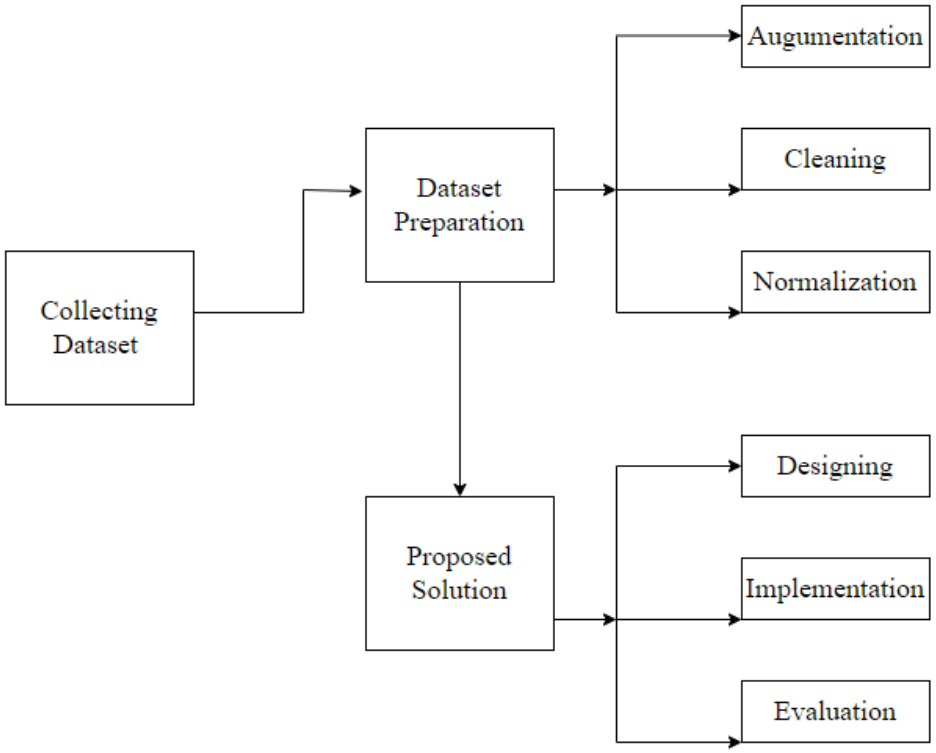


Figure 3.1: Overall Process.

3.2. Data Collection

To conduct training and testing the hybrid attention and relative average discriminator-based GAN for MR image reconstruction, there are two benchmarked MR Image datasets: CC-359 (Souza et al., 2018), and Stanford(Sawyer et al., 2013) used.

The brain datasets were obtained from Calgary Campinas brain MR raw data repository (Souza et al., 2018), containing 35 participants. Among them, 4250/5950 MRI brain slices were used for training, and the remaining were used for testing. The Calgary Campinas public brain magnetic resonance (MR) images dataset[1] was created through a joint project involving researchers from both the Vascular Imaging Lab at the University of Calgary and the Medical Image Computing Lab at the University of Campinas (UNICAMP).

The knee datasets were obtained from Stanford Fully Sampled 3D FSE Knees repository (Sawyer et al., 2013), containing 20 subjects. Among them, 1800/2000 slices were used for training, and the remaining were used for testing. The knee datasets were available on the Mridata[2] website which provides an open platform for researchers to share their magnetic resonance imaging (MRI) raw k-space datasets. The website is specifically created to make the sharing of MRI datasets from various vendors easier, by offering automatic conversion to the International Society for Magnetic Resonance in Medicine row data (ISMRMRD), parameter extraction, and thumbnail generation features.

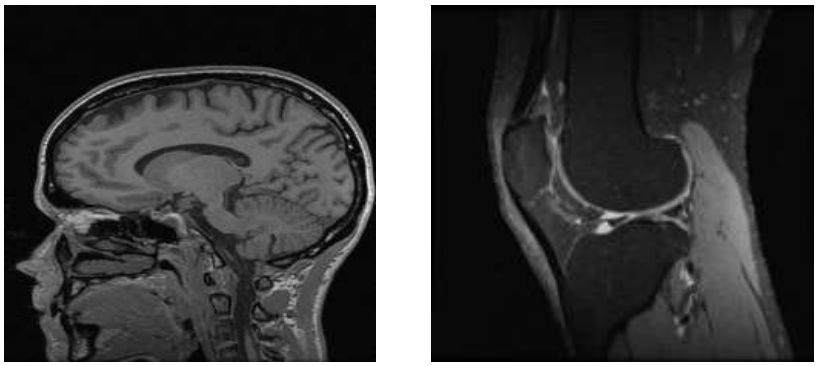


Figure 3.2: Brain and Knee MR image dataset sample.

[1] https://sites.google.com/view/calgary-campinas-dataset/home

[2] http://www.mridata.org/

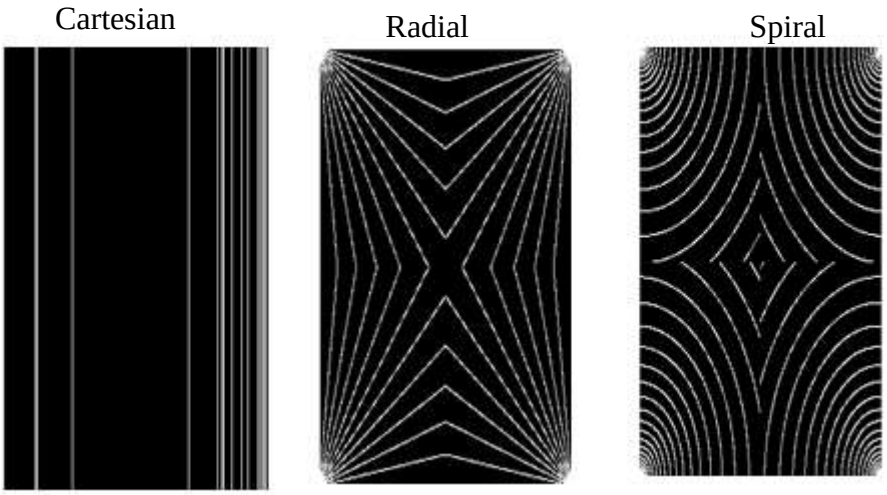


Figure 3.3: Cartesian and non-cartesian under sampling rate

The 12.5% and 10% imply that the under-sampled acquisition is eight times(8x) and 10 times (10x) faster compared to fully sampling the k-space data respectively.

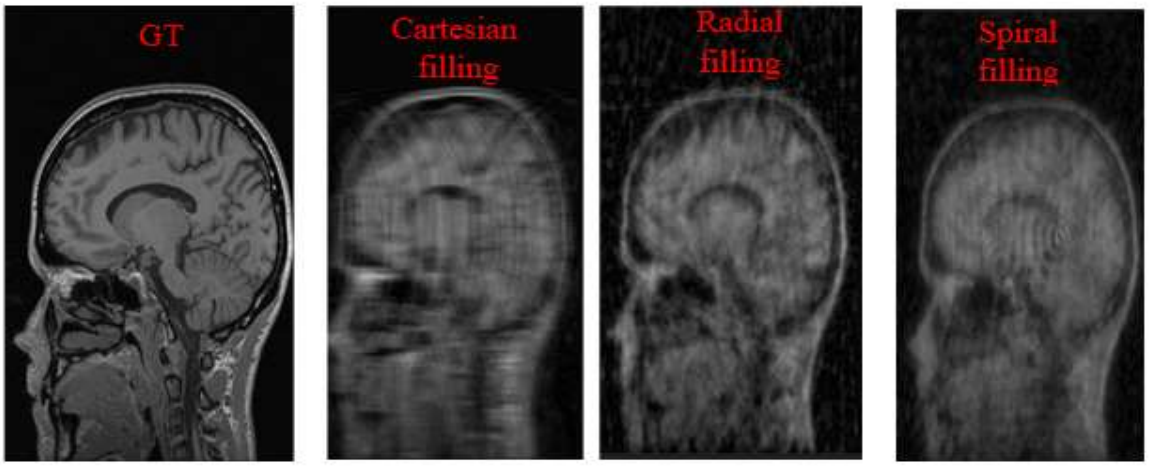


Figure 3.4: Fully sampled and under-sampled brain MR images

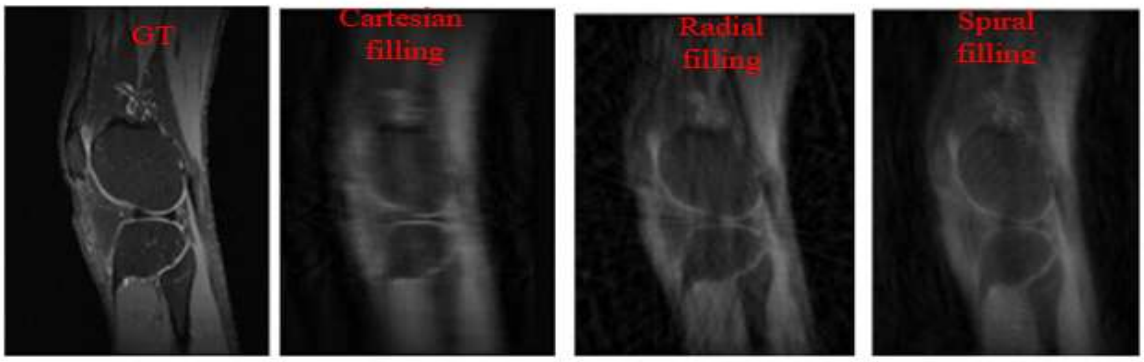


Figure 3.5: Fully sampled and under-sampled knee MR images

3.3. Pre-processing Techniques

3.3.1. Resizing Images

The matrix size of all the images was reshaped to 256 × 256. All fully-sampled k-space data would be retrospectively under-sampled using cartesian sampling with a sampling rate of 12.5% mask. We also used another two non-cartesian trajectories, spiral and radial with under-sampling rates of 10% mask.

3.3.2. Normalization Techniques

The normalization method is commonly used in computer vision tasks to improve the performance of deep neural networks. It helps to reduce the impact of illumination and contrast variations in the input images, making the network more robust and improving its accuracy. In our works, the input image is first normalized to have zero mean and unit standard deviation by subtracting the mean and dividing by the standard deviation. Then, it is scaled to the range of [-1,1] by subtracting the minimum value of the normalized image, multiplying by 2, and then dividing by the range of the normalized image. Finally, 1 is subtracted from this value to shift the range to [-1,1]. In this research, all images are linearly normalized to [-1,1] during training.

$$X = 2\left(\frac{\mathrm{y} - \mu}{\delta}\right) - 1 \qquad (3.1)$$

Where x is the normalized image, y is the input image that has a value between 0 and 1, and μ and δ are mean and variance respectively. Both mean and variance have 0.5 values.

3.3.3. Data Augmentation Transformations

A series of data augmentation transformations would be applied to the training images to increase the variability of the training data and improve the performance of our model. In our work, we used five times the number of data available for training and testing of brain and knee datasets by applying the following augmentation transformations.

Here are the details of the transformations:

- ✓ Random Horizontal Flip: We makes randomly flip the input image horizontally with a probability of 0.5.
- ✓ Random Vertical Flip: We makes randomly flip the input image vertically with a probability of 0.5.

- ✓ Random Rotation (90, fill=0): we make randomly rotate the input image by an angle between -90 degrees and 90 degrees. The fill parameter sets the value used to fill the empty pixels after the rotation. In this case, the empty pixels will be filled with zeros.
- ✓ Random Crop (244): We makes randomly crop the input image to a size of 244x244 pixels.
- ✓ Padding: We add a padding of 6 pixels to all sides of the input image.

The resulting transformed image must have the shape of (channels, height, and width), which is the expected format for input to a PyTorch neural network using a PyTorch tensor.

3.4. Development Tools

To design and implement the proposed work, various types of development tools were used in this study. These tools include prototype development platforms, UML modeling tools, and other research-related tools. The sections that follow provide a high-level overview of these development tools.

3.4.1. Design Tools

Design tools are mediums that can be used to create, present, and interpret design concepts. In the proposed technique, different hardware and software tools were used for designing purposes. It is a lightweight and powerful graphic design tool for producing professional-looking flowcharts, network diagrams, and flowchart diagrams, among other things.

3.4.2. Hardware Tools

The hardware tools listed below were used for the research implementation.

Table 3.1: Hardware Tools

No.	Tools	Used for
1.	RAM	Accelerating the training process cooperatively with GPU
2.	Hard Disk	Storing large datasets
3.	GPU	Increasing the computation and accelerating the training process

3.4.3. Software Tools

Various writing software and coding tools were used to implement the study via coding, and these are listed below with their descriptions.

Anaconda[3]:- is an open-source data science distribution of the Python and R programming language that attempts to simplify package management and deployment. Anaconda's package versions are handled by the conda package management system, which examines the current environment before completing an installation to prevent interrupting other frameworks and packages.

Jupyter Notebook[4]:- is an open-source web application that allows users to create and share documents that contain live code, equations, visualizations, and narrative text. One of the main features of Jupyter Notebook is the ability to run code interactively. Users can execute individual code cells and see the output immediately, making it an excellent tool for exploratory data analysis, prototyping, and teaching.

PyTorch[5]: is a popular open-source machine learning library that is primarily used for building deep neural networks. It was developed by Facebook's AI Research (FAIR) team and provides support for both CPU and GPU computing. PyTorch is known for its ease of use and dynamic computational graph, which makes it a popular choice for researchers and developers.

TensorBoard[6]: is a popular open-source visualization tool in machine learning. It allows users to visualize the training process of their machine learning models in real time, as well as inspect the architecture of the model, display histograms of weights and biases, and view embeddings in 3D. However, it can also be used in conjunction with PyTorch by logging data and metrics from the PyTorch training process into TensorBoard.

---

[3] https://anaconda.org/

[4] https://jupyter.org/

[5] https://pytorch.org/

[6] https://pypi.org/project/tensorboard/

3.5. Feature Extraction Network

We used Residual U-Net (Z. Zhang et al., 2018) as a feature extractor for MR image reconstruction using GAN. In MR image reconstruction, the goal is to reconstruct a high-quality image from under-sampled k-space data. One way to use Residual U-Net for MR image reconstruction is to use it as a feature extractor for a deep learning model that takes under-sampled k-space data as input and produces a high-quality image as output. The ResNet (He et al., 2016) model can be used to extract high-level features from the under-sampled k-space data, which can then be used as input to the image reconstruction model.

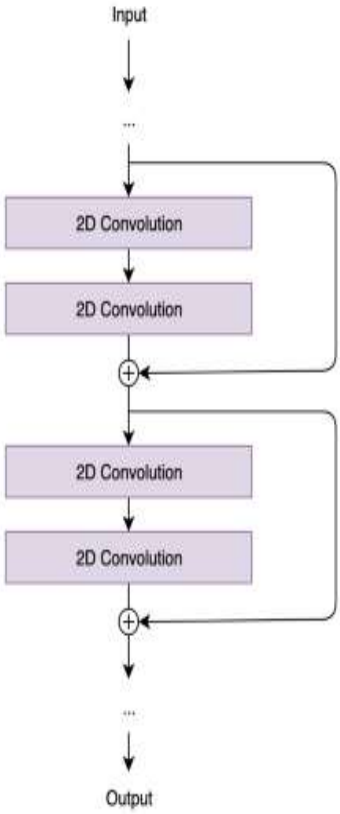


Figure 3.6: ResNet architecture. Cited from (He et al., 2016)

These skip connections act as gradient superhighways, allowing the gradient to flow unrestricted. The gradient becomes smaller and smaller, making the updates to the initial layers very small, increasing the training time considerably. It solves the vanishing gradient problem by making the local gradient became 1. So, as this gradient is backpropagated, it does not decrease in value because the local gradient is 1.

In our works, first, we build the feature of the Residual U-Net model and define an input layer for the image reconstruction model that takes under-sampled k-space data as input. We then use the ResNet model features to extract high-level features from the under-sampled k-

space data and pass these features through several layers of convolutional transposed layers to generate the reconstructed image. We then define the image reconstruction model with an optimizer and loss function. Finally, we train the model using under-sampled k-space data and corresponding high-quality images.

3.6. Evaluation Metrics

In the proposed method, we used the PSNR, SSIM, and NMSE values and subjective evaluation metrics like a human evaluation to evaluate the performance of the reconstruction results. In MR image reconstruction, the NMSE is often used in conjunction with other metrics such as the peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM) to evaluate the quality of the reconstructed images. These metrics can provide complementary information and help to give a more complete picture of the reconstruction quality. In addition to that, we used the rank-sum test method to test the statistically significant difference between each method. The $P<0.05$ indicates statistical significance.

3.6.1. NMSE (Normalized Mean Square Error)

Normalized Mean Square Error is a commonly used metric for evaluating the quality of image reconstruction in magnetic resonance imaging (MRI). The NMSE measures the discrepancy between the reconstructed image and the ground truth image, normalized by the energy of the ground truth image. A lower NMSE value indicates a better reconstruction quality, as it indicates that the reconstructed image is closer to the ground truth image.

NMSE is most easily defined via the mean squared error (MSE). Mean Squared Error (MSE) between two images such as ground truth g (x, y) and reconstructed image gˆ (x, y) in sizes of MN is defined as

$$MSE = \frac{1}{MN}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[\hat{g}(i,j) - g(i,j)].^2 \qquad (3.2)$$

$$NMSE = \frac{MSE}{Var(y)} \qquad (3.3)$$

Where MSE is the Mean Squared Error, and Var(y) is the variance of the true values of the target variable.

The variance is calculated as the average of the squared differences between each data point and the mean of the target variable. The formula for calculating the variance of a sample of data is:

$$Var(y) = \frac{1}{n}\sum(y_i - \hat{y})^2 \qquad (3.4)$$

Where n is the number of data points, y_i is the i-th data point, $\bar{y}$ is the mean of the target variable, and Σ represents the sum of the squared differences.

3.6.2. PSNR (Peak Signal to Noise Ratio)

PSNR is a commonly used metric for evaluating the quality of image reconstruction in MRI. PSNR measures the ratio of the maximum possible pixel value to the mean squared error (MSE) between the reconstructed image and the ground truth image. The PSNR metric provides a quantitative measure of the reconstruction quality, with higher PSNR values indicating better image quality.

PSNR is expressed as:

$$PSNR = 20log_{10}\left(\frac{MAXf}{\sqrt{\text{MSE}}}\right) \qquad (3.5)$$

Here, MAXf is the maximal in the image data.

3.6.3. Structural Similarity Index Measure (SSIM)

SSIM is a useful metric for evaluating the quality of MR image reconstruction because it takes into account both low-level image features (such as contrast and luminance) and high-level structural information (such as edges and texture). SSIM is a metric used to measure the similarity between two given images. It tries to look at a group of pixels to better determine if the two images are different or not.

In the SSIM index method, a quality measurement metric is calculated based on the computation of three major aspects termed luminance, contrast, and structural or correlation term. This index is a combination of the multiplication of these three aspects (Sara et al., 2019).

Again luminance, contrast, and structure of an image can be expressed separately as:

$$L(x,y) = \frac{2\mu_x \mu_y + c1}{\mu_x^2 + \mu_y^2 + c1} \quad (3.6)$$

Where c1 = $(k1 + l)^2$ is the non-negative regularization constant used to avoid instability when the mean is close to zero, l is the dynamic range value, and k1 is the small constant value.

$$C(x,y) = \frac{2\delta xy + c2}{\delta_x^2 + \delta_y^2 + c2} \quad (3.7)$$

Where c2 = $(k2 + l)^2$ is the non-negative regularization constant used to avoid instability when the mean is close to zero, l is the dynamic range value, and k2 is the small constant value.

$$S(x,y) = \frac{\delta_{xy} + c3}{\delta_x \delta_y + c3} \quad (3.8)$$

Where, μx average mean of x, μy average mean of y, δ_x^2 variance of x, δ_y^2 the variance of y and δ_{xy} covariance of x and y, and c3 = c2/2

In this context, L represents luminance, which is utilized to compare the brightness levels between two images. C signifies contrast, which is employed to distinguish the range between the brightest and darkest regions of two images. S denotes structure, which is used to compare the local luminance pattern between two images and identify similarities and dissimilarities. Additionally, α, β, and γ are positive constants (Sara et al., 2019).

The structural Similarity Index Method can be expressed through these three terms:

$$SSIM(x,y) = [L(x,y)]^{\alpha} \cdot [C(x,y)]^{\beta} \cdot [S(x,y)]^{\gamma} \quad (3.9)$$

Where α, β, and γ are parameters that are used to adjust the relative importance of the three Components

The SSIM index ranges from -1 to 1, with 1 indicating perfect similarity between the reconstructed image and the ground truth image, and -1 indicating complete dissimilarity. In practice, SSIM values typically range from 0 to 1, with higher values indicating better reconstruction quality.

CHAPTER FOUR

4. PROPOSED SOLUTION

4.1. Chapter Overview

This chapter describes the proposed solution architecture with mathematical expressions and its necessary diagrammatic representation of the MRI reconstruction. The proposed solution of reconstructing an MR image based on a given K-space data at different under-sampling rates is presented in this chapter. The reconstructed MR image should have a noise-free, better-quality visualization and the MR image details information of its original image. Accordingly, it mainly introduces the model architecture used to reconstruct the MR image from the human brain and knee K-space data. And also, problem definition and notations, the different optimizing loss functions, and training schemes are described in detail.

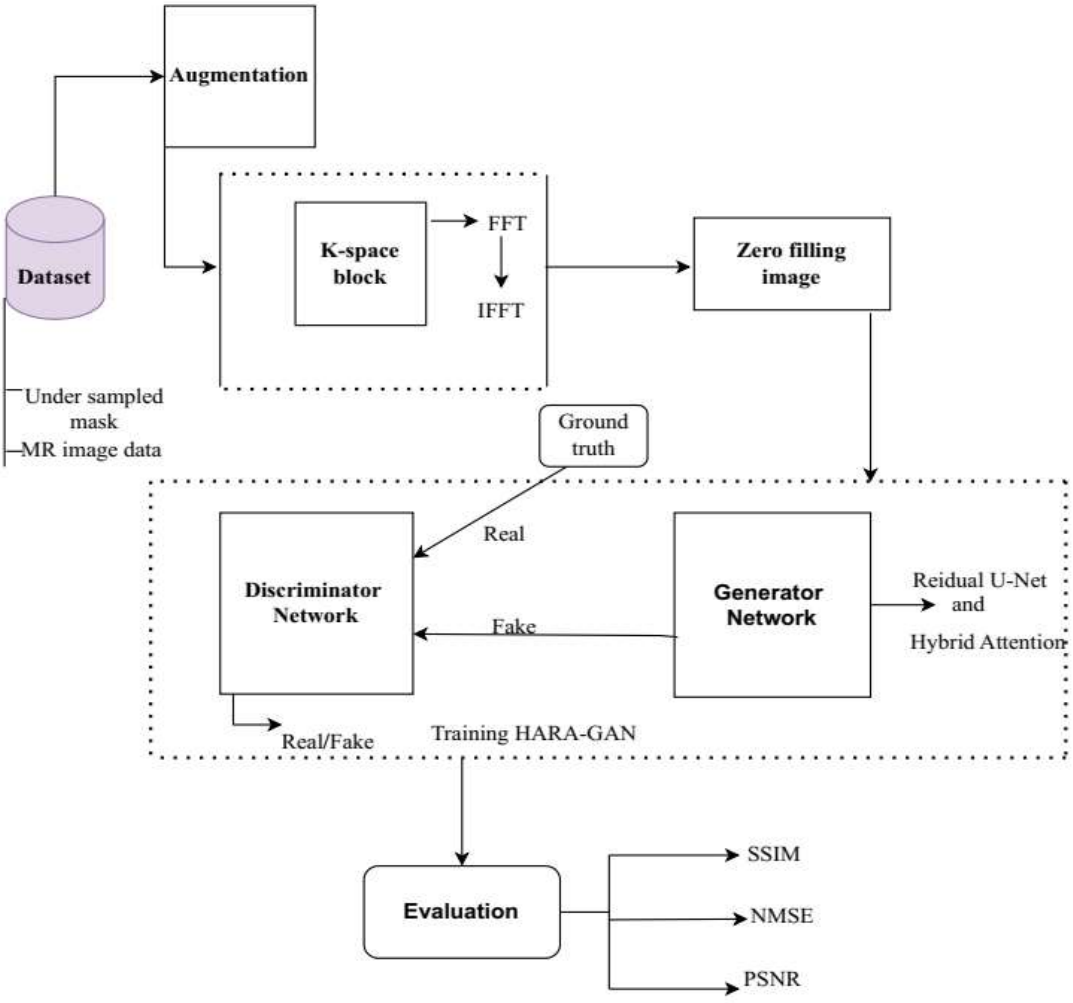


Figure 4.1: The proposed model process flow

Here are the steps our works follow to reconstruct MR images:

Step 1: Data Preparation

We create two directories, one for the input images in png format and another for the masks in .tiff format. Then we read the files from these directories and preprocess the data by resizing and normalizing the images.

The K-space under-sampling block should have an operation to create the under-sampled image from the mask and fully sampled data. For creating zero-filling images using Fast Fourier transform (FFT) and inverse Fast Fourier transform (IFFT), we used the NumPy module. First, apply FFT to the input image, then replace the high-frequency coefficients with zeros and apply IFFT to the result to get the zero-filled image.

Step 2: Creating the Generator and Discriminator Models

We used the PyTorch and Torchvision libraries to create the generator and discriminator models. The generator was taking the input image and generating results, while the discriminators were differentiating between the real and fake images.

Step 3: Loss Functions

We used cyclic data consistency loss and WGAN loss functions to ensure that the generated images could accurately reconstruct the original image.

Step 4: Training

Train the generator and discriminator models with the zero-filled image to reconstruct the MR image from different under-sampled rates.

Step 5: Testing

Use the trained generator model to generate the under-sampling for the input images. Then use the reverse operation to get the reconstructed images from the under-sampled data.

Step 6: Evaluation

Finally, we evaluate the performance of our model by calculating the Peak Signal-to-Noise Ratio (PSNR), Normalized Mean Squares Error (NMSE), and Structural Similarity Index Measure (SSIM) between the reconstructed and original images.

4.2. Overview of Proposed Model Architecture

The proposed approach, entitled "HARA-GAN," is used to reconstruct MR images from low under-sampled k-space data. It mainly depends on the Generative Adversarial Network (Creswell et al., 2018). In generative adversarial networks, there are generator and discriminator modules. In the generator module, our proposed method used two residual U-net(Ronneberger et al., 2015). And also again, the U-net has encoder and decoder modules. In the encoder part, we used the SSA (spatial self-attention) (R, 2021) to encode contextual information into local features and improve intra-class representation. On the decoder side also we used SCA (spatial channel-wise attention) (G. Li et al., 2021) to decode the encoded features based on the weights of each feature. The generator generates a fake image by inputting information into a real image, which is used to fool the discriminator.

In discriminator modules, we used a relative average discriminator (Jolicoeur-Martineau, 2019) to improve the discriminator performance. The work of the discriminator is to judge the real or fake images generated by the generator. For the image generated by the generator, the result of the discriminator is fake. For the real image, the result of the discriminator is real. The generator is constantly learning to generate images infinitely close to the real image, and the discriminator is constantly optimized to judge the image generated by the generator.

In the context of generative adversarial networks, the term cyclic data consistency loss(Zhu et al., 2017) with WGAN loss(Arjovsky et al., 2017)refers to a specific combination of loss functions used for training GAN models. Thus, in the process of continuous generation and discrimination, the cyclic data consistency loss(Zhu et al., 2017) with WGAN loss(Arjovsky et al., 2017) function is generated, and the network is continuously optimized by generating those losses of function. By combining these two loss functions, the model is trained with both the WGAN loss, which encourages realistic output generation, and the cyclic data consistency loss, which enforces consistency between the original and reconstructed images. This combined loss enables the model to generate realistic and coherent translations while maintaining content consistency across domains.

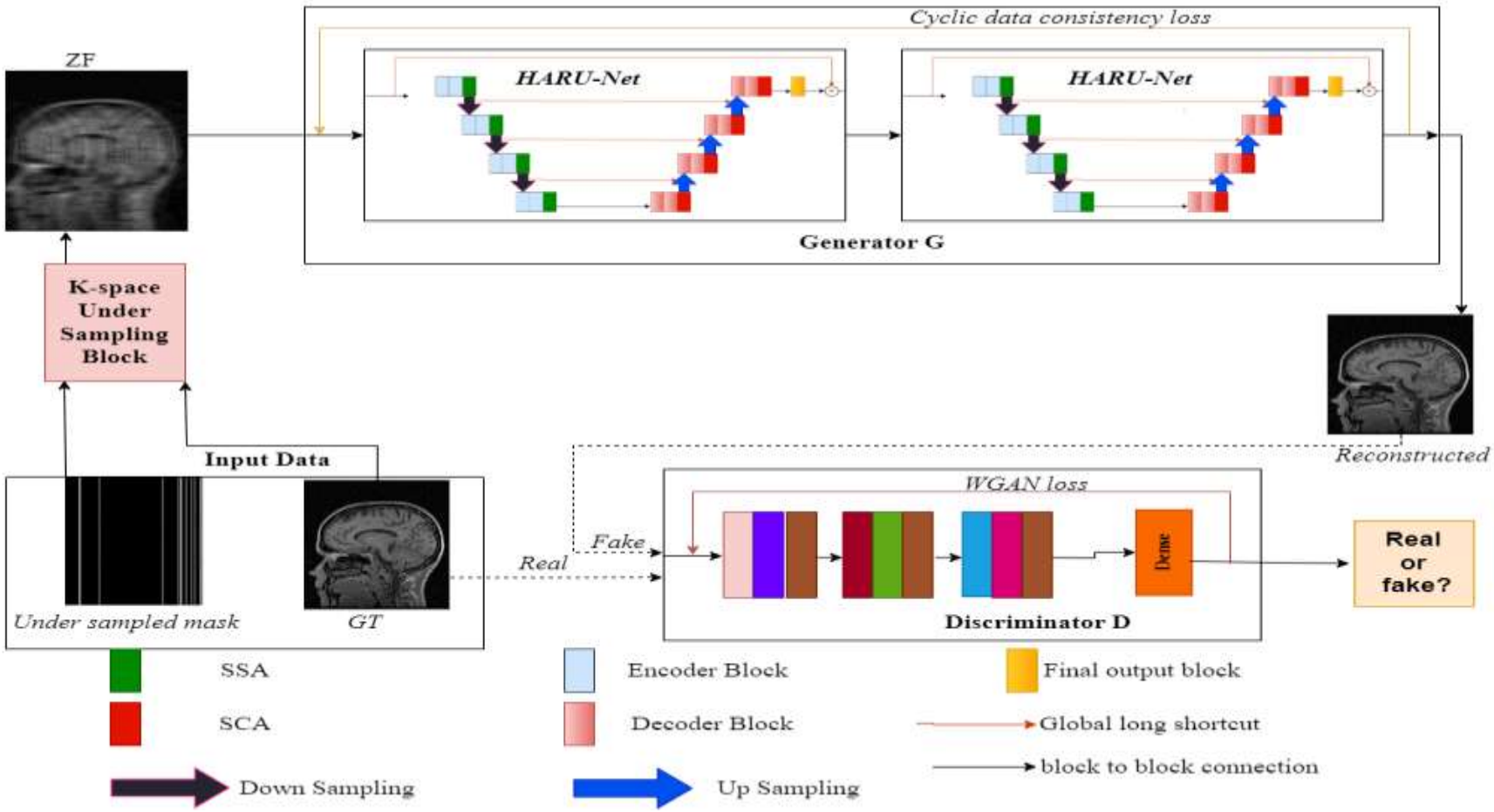


Figure 4.2: General proposed solution architectures.

4.3. K-Space Under-sampling Blocks Definition

The overall architectures of our method are illustrated in Figure 4.2. We designed a k-space under-sampled (KU) block as shown in Figure 4.3 with the ground truth image SGT and the under-sampling mask R of the module as inputs and the zero-filling image SZF as the output. In the KU block, there are two sub-modules called the Fourier to transform R (FR) block and the inverse Fourier transform R (IFR) block. The FR block employs the Fourier transform to convert the GT image from the time domain to the frequency domain and uses the R mask to retrieve the under-sampled k-space samples. The inverse Fourier transform of the IFR block is then used to convert the under-sampled frequency domain image into the time domain and produce the zero-filling image SZF.

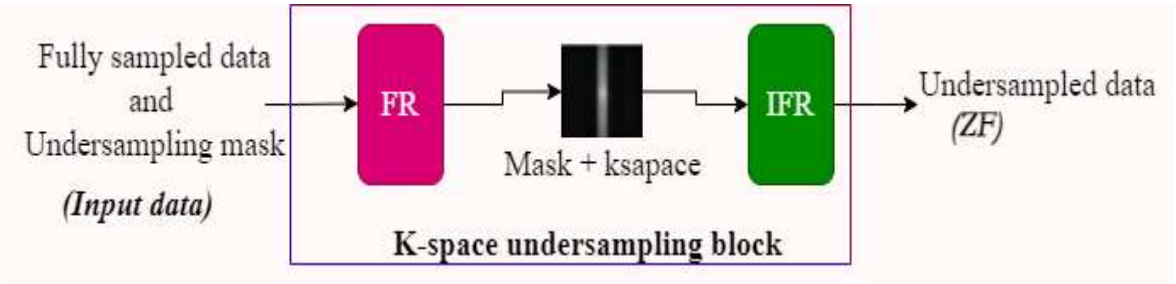


Figure 4.3: K-Space under-sampled block

The IFR block is represented as follows

$$IFR = F * (FR) = F * (F(SGT).R) \qquad (4.1)$$

Where FR represents the formula of the FR block, F stands for Fourier transform, and IF represents the inverse Fourier transform.

We denote the under-sampled raw MRI data (k-space measurements) using the sampling mask R as M. Then, its zero-filling reconstruction Zf can be obtained by the following equation:

$$Zf = FHRH\,(M) \qquad (4.2)$$

Where F is the Fourier operator, and superscript H indicates the conjugated transpose of a given operator. Similarly, turning any image S_i into its under-sampled measurement MS_i with the given sampling mask R can be done via the inverse of the reconstruction process:

$$MSi = RF\,(Si) \qquad (4.3)$$

4.4. Generator Architecture

The generator models, as seen in Figure 4.2, are made up of two HARU-Net (hybrid attention residual U-Net) modules, each containing long-skip connected residual blocks, encoding and decoding blocks, SSA, and SCA. SSA is applied in encoding blocks of the down-sampling process and SCA is applied after the decoding block in the up-sampling process.

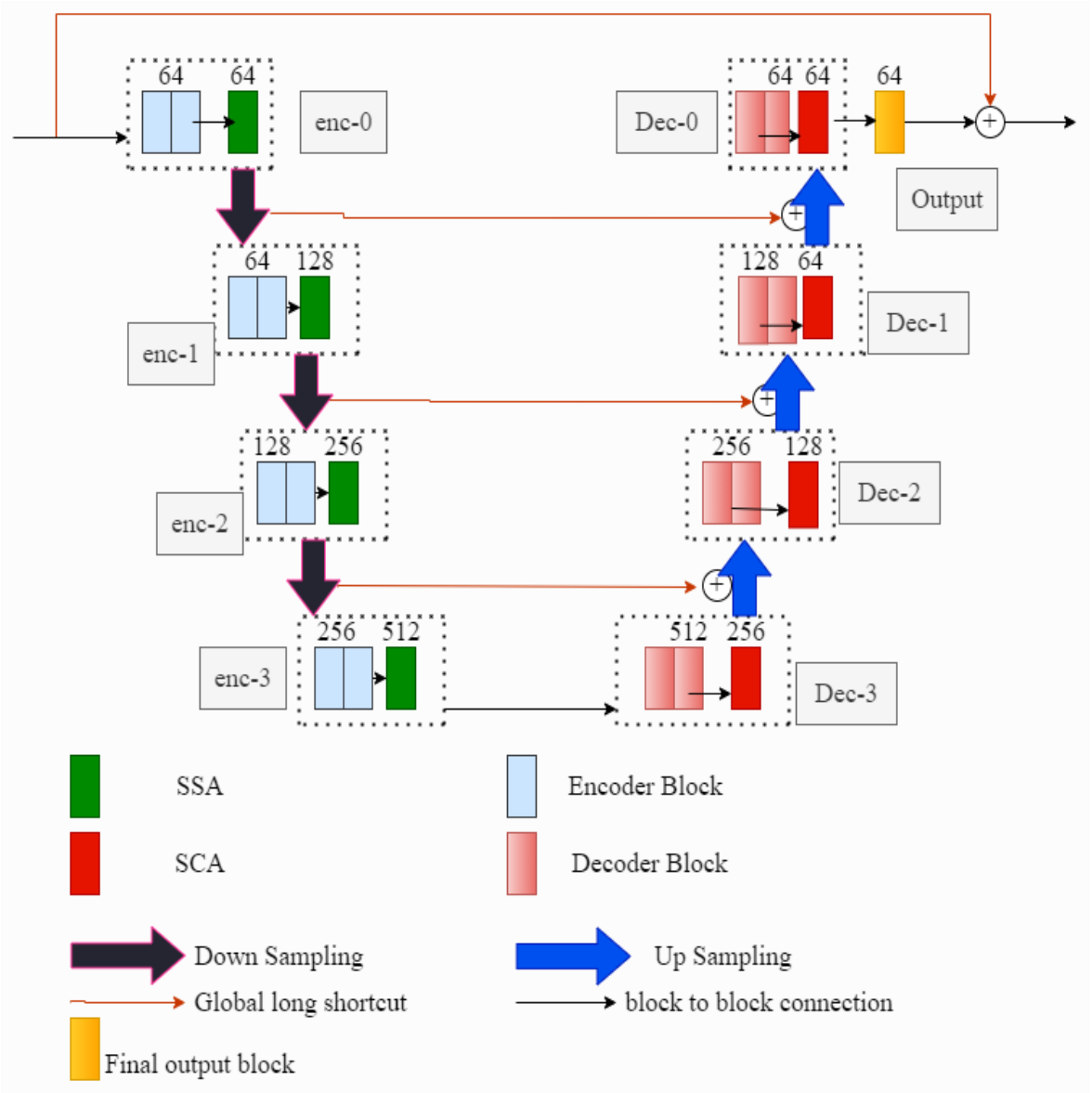


Figure 4.4: Architectures of Hybrid Attention Residual U-Net Generator

Figure 4.4 shows that each encoding and decoding block comprises two short-skip connection residual blocks. HARU-Net has four encoding and four decoding blocks and a 2D convolution layer, with light blue representing the encoding block and pink representing the decoding block.

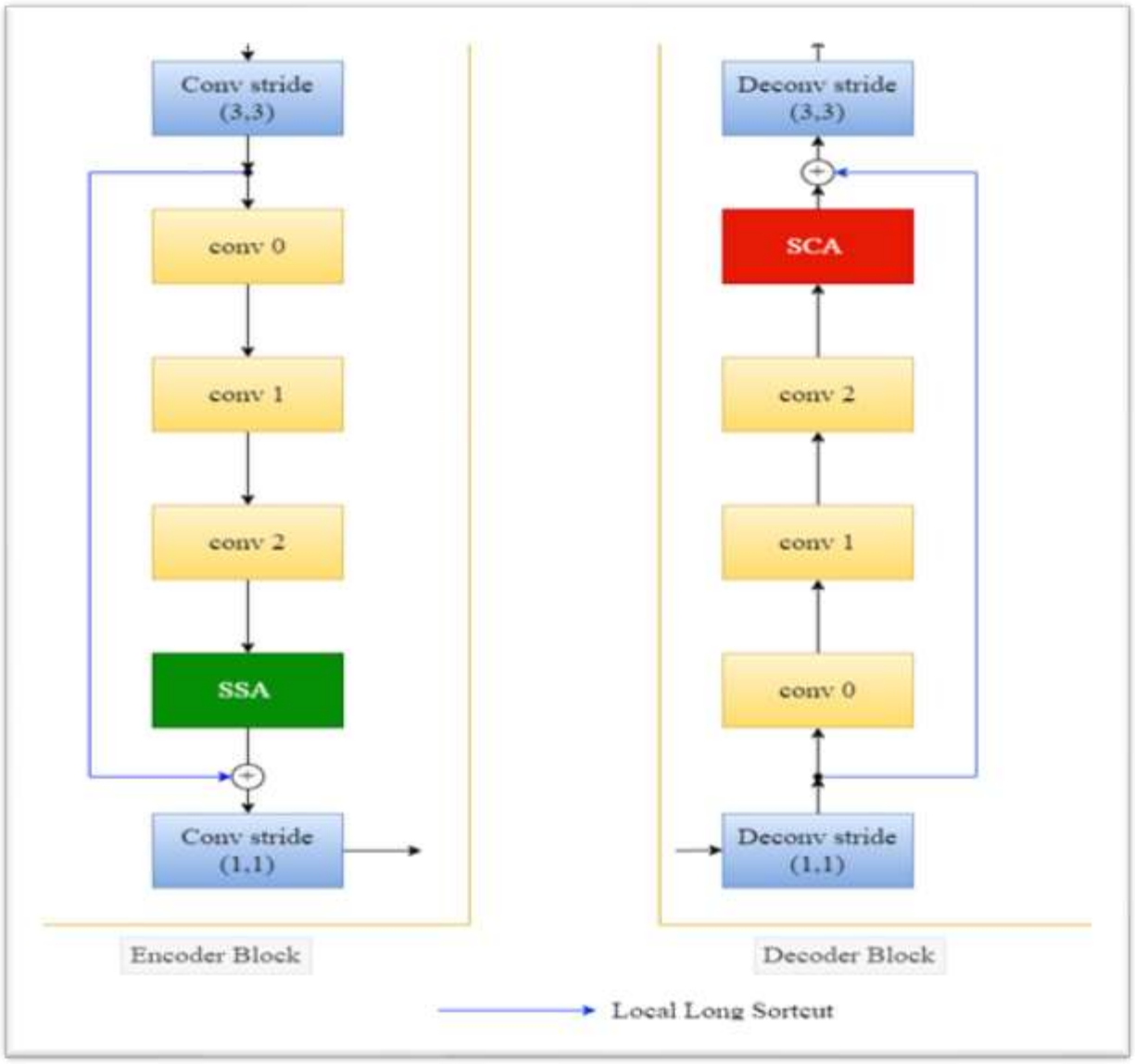


Figure 4.5: Architectures of internal Residual encoder and decoder block

Figure 4.5 shows the residual encoder and residual decoder in each block. The 4D tensor is used as input, and the 2D convolution layer has a filter size of 3×3 and a stride of 3. A global long skip connection connects the encoder and decoder, and each local short skip connection residual block, colored blue line, increases the network's depth.

The encoder block comprises three convolutional layers (conv0, conv1, and conv2) and an SSA block. Conv0 is a convolution with a stride of 1 and a filter size of 3. Conv0, conv1, and conv2 have the same stride and filter size and used the LeakyReLU activation function. The number of feature maps in conv0 and conv2 has the same, but conv1 has different feature maps. The output of conv2 serves as the input for the SSA block.

And also, the decoder block comprises three deconvolutional layers (deconv0, deconv1, and deconv2) and an SCA block. Deconv0 is a convolution with a stride of 1 and a filter size of 3. deconv0, deconv1, and deconv2 have the same stride and filter size. The number of feature maps in deconv0 and deconv2 has the same, but deconv1 has different feature maps. The output of deconv2 serves as the input for the SCA block. The activation function used in SSA and SCA blocks was sigmoid. The LeakyReLU activation function was used in each encoder and decoder convolutional block.

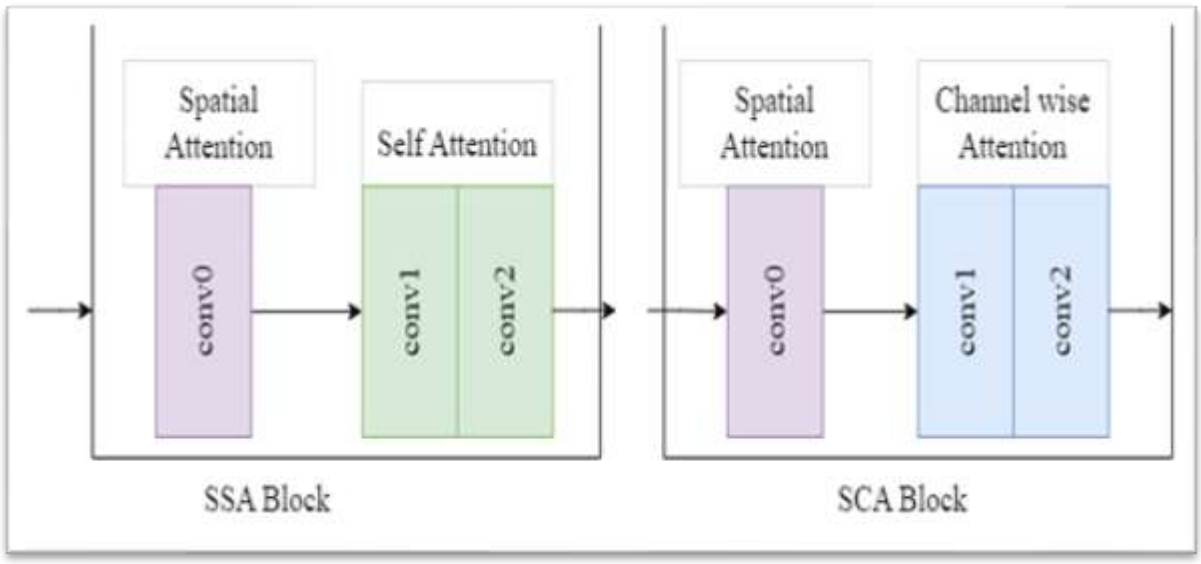


Figure 4.6: SSA and SCA block architectures

We have also introduced the structure of the SSA and SCA block. The encoding blocks utilize a down-sampling method, while the decoding blocks use an up-sampling method. Following each layer of encoding blocks, the SSA block is used to extract image details with intra information, and the SCA block is used to extract image details and give weights following each layer of decoding blocks. The input of each SCA block comes from the decoding block of the same layer, and the output serves as the input for the subsequent block. After the final SCA block, a 2D convolution with a stride of 1 is used to obtain an output image of the same size as the input image, using the same mode in the convolution operation. Our method employs a dual cascade generator, although one generator is capable of completing end-to-end operations, from zero-filling MRI image to the final reconstructed image. To obtain the final reconstruction, we add the zero-filling input to the output from the generator G, which is similar to other current deep learning-based CS-MRI methods(Q. Huang et al., 2019; G. Li et al., 2021; Quan et al., 2018).

4.5. Hybrid Attention and Residual U-Net Mechanism

HARU-Net (Hybrid Attention Residual U-Net) is a deep learning model used for magnetic resonance imaging (MRI) reconstruction. It is a modified version of the standard Residual U-Net architecture that incorporates both spatial self-attention (SSA) and spatial channel-wise attention (SCA) modules. Spatial self-attention and spatial channel-wise attention are two attention mechanisms used in MRI reconstruction in Residual U-Net that offer several advantages. The SSA module is added to the encoder part of the network, while the SCA module is added to the decoder part. The encoder extracts feature from the input image, and the SSA module selectively emphasizes informative regions while suppressing noise and artifacts. We used Spatial self-attention to capture long-range dependencies within the image by learning which spatial locations are most relevant for predicting each pixel in the output. This mechanism helps the network to focus on important image features while ignoring irrelevant ones, which can improve the overall performance of the network in capturing fine details in the reconstructed images.

The decoder then uses the extracted features to generate the output image, and the SCA module selectively enhances important channels while suppressing irrelevant ones. Spatial channel-wise attention, on the other hand, helps the network to selectively amplify or suppress specific feature maps in each layer of the network based on their importance for the reconstruction task. By doing so, the network can assign higher weights to important channels and suppress less useful ones, leading to improved reconstruction performance and better handling of noise and artifacts in the input image.

The hybrid attention mechanism in HARU-Net thus helps to improve the quality and consistency of the reconstructed MRI images, as well as reduce noise and artifacts. Additionally, HARU-Net also uses residual connections, which allow the network to learn the difference between the input and output images, making it easier for the network to learn the mapping between the input and output. Generally, HARU-Net is a powerful and effective generator model for MRI reconstruction and has shown promising results for MR image reconstruction. The advantages of using spatial self-attention and spatial channel-wise attention in MRI reconstruction in Residual U-Net include better preservation of fine details, improved noise reduction, and better handling of artifacts and other imperfections in the input image.

4.6. Loss functions

Determining the best loss function for MRI reconstruction using GANs depends on several factors such as the quality of the data, the specific MRI application, and the computational resources available (Abu-Srhan et al., 2022). We modify the loss function used for training our models with the generator and discriminator networks. In addition to the cyclic data consistency loss (Zhu et al., 2017), we add the WGAN loss. The WGAN loss(Arjovsky et al., 2017) is used to train the discriminator network to distinguish between real and fake images, while the cyclic data consistency loss is used to train the generator network to produce images that are consistent with the input data.

Both WGAN loss and cyclic data consistency loss with relative average discriminator have been shown to be effective in MRI reconstruction. WGAN loss has been shown to be effective in training stable GANs by minimizing the Wasserstein distance between the distribution of real and generated images. This approach has been used successfully in MR image reconstruction tasks, producing high-quality images with reduced noise and artifacts.

Cyclic data consistency loss is expressed as:

$$L_{cyc}(G,F) = \sum_{x \sim pdat(x)} [||F(G(x) - x||] + \sum_{y \sim pdat(y)} [||G(F(y) - y||] \qquad (4.4)$$

where F and G are the generators that map from domain X to Y and from Y to X, respectively, x is an image from domain F, and y is an image from domain G. The || || denotes the L1 or L2 norm, which measures the distance between the generated and original images.

The loss function for the WGAN is expressed as:

$$L_{WGAN} = E_{\{x \sim P_{gen}\}}[f(x)] - E_{\{x \sim P_{real}\}}[f(x)] \qquad (4.5)$$

where P_real and P_gen are the true and generated probability distributions, respectively, and f is the critic function.

We train our models HARA-GAN, using the combination of the WGAN loss and the cyclic data consistency loss. During training, the generator network is updated to minimize both losses simultaneously, while the discriminator network is updated to maximize the WGAN loss. Generally, we used both WGAN loss and cyclic data consistency loss into a HARA-GAN for MRI reconstruction to improve the quality of the reconstructed images and ensure that they are consistent with the ground truth image.

4.7. Discriminator Architecture

In discriminator parts of our model, we used the relative average discriminator with a generator network called HARU-Net (Hybrid Attention Residual U-Net), to train our model to generate high-quality MRI images and to improve its performance. The generator network generates a reconstructed image, which is fed to the relative average discriminator along with the ground truth image. The discriminator then provides feedback to the generator, helping it to generate better-quality images.

The relative average discriminator is a theory that proposes a new way of training generative adversarial networks (GANs) by using a different loss function for the discriminator(Yuan et al., 2020). It is also, a type of discriminator used in MRI reconstruction using deep learning models. The main goal of the discriminator is to differentiate between the reconstructed images and the ground truth images. We selected to use a relative average discriminator over a traditional discriminator because it is designed to address the problem of over-smoothing in traditional adversarial networks used in MRI reconstruction, which can lead to the loss of fine details in the reconstructed images.

The idea is to make the discriminator estimate the probability that a given sample is more realistic than another sample on average, rather than estimating the absolute probability of being real or fake. This way, the discriminator can make use of the prior knowledge that half of its input data is true and half is fake, and avoid some of the problems of traditional discriminators such as mode collapse and vanishing gradients.

The advantage of using a relative average discriminator over a traditional discriminator is that it could improve the stability and performance of GAN training, and produce more diverse and realistic samples. Relative average discriminator uses a novel loss function that compares the relative differences between the reconstructed image and the ground truth image, rather than simply comparing pixel values as in traditional discriminators. This loss function is calculated using a relative average operation, which computes the ratio of the mean absolute difference between the images to the average pixel intensity of the ground truth image.

Generally, we used the relative average discriminator for MR image reconstruction using Generative adversarial networks which is an effective approach to improve the quality and consistency of the reconstructed images.

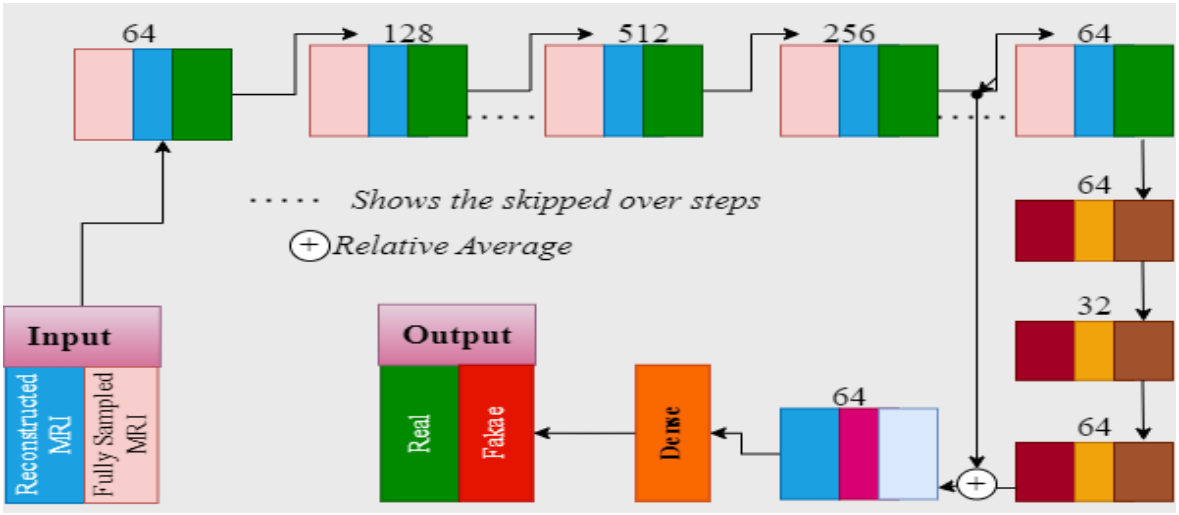


Figure 4.7: Relative average discriminator architectures

4.8. Training Pseudocode

Algorithm1: Proposed hybrid attention-G and relative Average-D GAN algorithm

1. **Initialize** *all the parameters of generators G and discriminator D*
   SSA *at the encoder and SCA at the decoder parts of residual U-net*
   Relative average discriminator *at discriminator sides*
2. **Input:** *Brain and knee MRI images A (1……n) with under sampling mask R(m) as* MS_i
3. ***For*** *numbers of epochs* ***do***
4. ***For*** *n steps* ***do***
5. ***Update*** *the discriminator D using*

$$\nabla_{\theta d}\frac{1}{A}\sum_{a=1}^{n}[\,\boldsymbol{E}_{\{x\sim \boldsymbol{P}_{gen}\}}[\boldsymbol{f}(\boldsymbol{x})]-\boldsymbol{E}_{\{x\sim \boldsymbol{P}_{real}\}}[\boldsymbol{f}(\boldsymbol{x})]\,]$$

6. ***Update*** *the generator G using*

$$\nabla_{\theta g}\frac{1}{A}\sum_{a=1}^{n}[\sum_{x\sim pdat(x)}[||F(G(x)-x||]+\sum_{y\sim pdat(y)}[||G(F(y)-y||]\;\;]$$

7. ***End for***
8. ***Evaluation*** *for validation data based on early stopping strategy*
9. ***Testing*** *using the MRI testing data for model evaluation*
10. ***End for***
11. **Output** *Reconstructed MR image* $\widehat{MSi}$

Bold, shows the updated components.

CHAPTER FIVE

5. IMPLEMENTATION OF THE PROPOSED SOLUTION

5.1. Chapter Overview

This chapter describes the implementation of the proposed model, which involves various tasks such as environment setup, data preprocessing, data augmentation, and the HARA-GAN training process. Furthermore, in this chapter, the experimental class is discussed as part of the implementation process. Moreover, it includes explanations of code sample snippets to aid in the implementation process.

5.2. Working Environments

This section describes the environmental setup and the specifications of hardware tools used for implementing the model. The working environments for MR image reconstruction using hybrid attention and relative average discriminator-based GAN are important aspects of the proposed solution. This includes the hardware and software used for the study. The hardware used includes a computer with a high-performance CPU and GPU for efficient processing of data. The following environments enable efficient development, training, and evaluation of the proposed solution for MR image reconstruction using hybrid attention and relative average discriminator-based GAN.

- ✓ Laptop:
  - Operating System: Windows 10
  - Processor: Intel® Core™ i5-8700 CPU @ 3.20GHz 3.19 GHz
  - Installed memory (RAM): 8.00 GB
  - System Type: 64-bit operating system, x64-based processor
- ✓ Desktop:
  - Operating system: Windows 10
  - Processor: Intel® Core™ i5-4590 CPU @ 3.30 GHz
  - Installed memory (RAM): 8.00 GB
  - System type: 64-bit operating system, x64-based processor
- ✓ Graphics processing unit (GPU):
  - Graphics: Intel ® HD Graphics 520/ NVIDIA Corporation
  - GeForce RTX 3060 with 12 GB RAM

5.3. Environment Setup

To carry out this study, a range of software and Integrated Development Environments (IDEs) that supports popular machine learning framework like TensorBoard and PyTorch is employed.

Anaconda: Anaconda is one of the working environments utilized in MR image reconstruction using hybrid attention and relative average discriminator-based GAN. It is an open-source tool used for simplified package administration and is commonly used to set up Python and its various modules. Anaconda version 1.9.7 is used in the implementation of the proposed solution to install all necessary libraries and modules.

Jupyter Notebook: is another working environment utilized in MR image reconstruction using hybrid attention and relative average discriminator-based GAN. It is an open-source tool that integrates software code, explanatory text, computational output, and multimedia resources into a single page. Jupyter Notebook is used in the project for code development and experimentation. It enables the researcher to easily write, test, and modify code, as well as document the process through the integration of text and multimedia resources. This makes Jupyter Notebook an ideal tool for developing and testing the proposed solution for MR image reconstruction using hybrid attention and relative average discriminator-based GAN.

TensorBoard: is used as a working environment for MR image reconstruction using hybrid attention and relative average discriminator-based GAN. It is a visualization toolkit for neural network training based on TensorFlow and provides useful features for the training process. TensorBoards's integration with TensorFlow allows for efficient development and training of the proposed solution for MR image reconstruction using GAN. The toolkit enables visualization and analysis of the training process, making it easier to identify and address any issues that may arise during training. Overall, TensorBoard is a valuable working environment for developing and training deep-learning models for MR image reconstruction.

To provide a comprehensive understanding of the work done in the proposed solution for MR image reconstruction using hybrid attention and relative average discriminator-based GAN, a detailed description of each stage is provided.

5.4. Dataset Generation Implementation

We used MR images and an under-sampling mask for training and testing our model. To do that we used the following code.

```
class MyData(Dataset):
def __init__(self, imageDir, maskDir, img_size=256,
is_training=True):
        super(MyData, self).__init__()
        images = glob.glob(join(imageDir, '*'))
        self.images = sorted(images)
        masks = glob.glob(join(maskDir, '*'))
        self.masks = sorted(masks)
        self.is_training = is_training
        self.img_size = img_size
        self.len = len(self.images)
```

Sample code 5.1: Dataset generation implementation

5.5. Data Augmentation Implementation

As previously mentioned, this work uses a series of data augmentation transformations that would be applied to the training images to increase the variability of the training data and improve the performance of our model.

```
def transform(self, img_A, img_B):
        totensor = ToTensor()  # rescale to [0,1.]
        for i, img in enumerate([img_A, img_B]):
            img = totensor(img)
            img = random_resize_crop(img)
            img = random_affine(img)
            img = random_h_flip(img)
            img = random_v_flip(img)
            img = random_perspective(img)
            if i == 0:
                img_A = img
            else:
                img_B = img
        return img_A, img_B
```

Sample code 5.2: Dataset augmentation implementation

5.6. HARA-GAN for MR Image Reconstruction

The proposed HARA-GAN model comprises two residual U-nets generators, which utilize hybrid attention mechanisms and relative average discriminator networks. The generator network is composed of two Residual U-net generators, which include a residual encoder block and a residual decoder block, as explained in section 4.5. The encoders consist of 12-layer neural networks with spatial self-attention, while the decoders have 13-layer neural

networks with spatial channel attention, ranging from 64 to 512 filters for computational efficiency, followed by a ReLU activation function. There are four residual skip connections as (G. Li et al., 2021).

```
class Generator(nn.Module):
    def __init__(self):
        super(Generator, self).__init__()
    def forward(self, x):
        x_e0 = self.e0(x)
        x_e1 = self.e1(x_e0)
        x_e2 = self.e2(x_e1)
        x_e3 = self.e3(x_e2)
        x_d3 = self.d3(x_e3)
        x_d2 = self.d2(x_d3 + x_e2)
        x_d1 = self.d1(x_d2 + x_e1)
        x_d0 = self.d0(x_d1 + x_e0)
        out = self.dd(x_d0)
        return out + x
```

Sample code 5.3 Generator function implementation

Table 5.1: Content of the generator encoder architecture

Layers	**Content of each block**
1	Convolution - (filters = 64, kernel size= (3, 3), stride= (2, 2), padding= (1, 1), bias=False), LeakyReLU(negative_slope=0.2)
2-3	Convolution - (filters = 64, kernel size= (1, 1), stride= (1, 1)), sigmoid ()
4	Convolution - (filters = 128, kernel size= (3, 3), stride= (2, 2), padding= (1, 1), bias=False), LeakyReLU(negative_slope=0.2)
5-6	Convolution - (filters = 128, kernel size= (1, 1), stride= (1, 1)), sigmoid ()
7	Convolution - (filters = 256, kernel size= (3, 3), stride= (2, 2), padding= (1, 1), bias=False), LeakyReLU(negative_slope=0.2)
8-9	Convolution - (filters = 256, kernel size= (1, 1), stride= (1, 1)), sigmoid ()
10	Convolution - (filters = 512, kernel size= (3, 3), stride= (2, 2), padding= (1, 1), bias=False), LeakyReLU(negative_slope=0.2)
11-12	Convolution - (filters = 512, kernel size= (1, 1), stride= (1, 1)), sigmoid ()

Table 5.2: Content of the generator decoder architecture

Layers	Content of each block
1	Deconvolution - (filters = 512, kernel size= (3, 3), stride= (2, 2), padding= (1, 1), bias=False), LeakyReLU(negative_slope=0.2)
2-3	Deconvolution - (filters = 256, kernel size= (1, 1), stride= (1, 1)), sigmoid ()
4	Deconvolution - (filters = 256, kernel size= (3, 3), stride= (2, 2), padding= (1, 1), bias=False), LeakyReLU(negative_slope=0.2)
5-6	Deconvolution - (filters = 128, kernel size= (1, 1), stride= (1, 1)), sigmoid ()
7	Deconvolution - (filters = 128, kernel size= (3, 3), stride= (2, 2), padding= (1, 1), bias=False), LeakyReLU(negative_slope=0.2)
8-9	Deconvolution - (filters = 64, kernel size= (1, 1), stride= (1, 1)), sigmoid ()
10	Deconvolution - (filters = 64, kernel size= (3, 3), stride= (2, 2), padding= (1, 1), bias=False), LeakyReLU(negative_slope=0.2)
11-12	Deconvolution - (filters = 64, kernel size= (1, 1), stride= (1, 1)), sigmoid ()
13	Convolution - (filters = 64, kernel size= (3, 3), stride= (1, 1), padding= (1, 1)), instance normalization, Tanh ()

5.6.1. Spatial Attention Implementation

```
class SpatialAttention(nn.Module):
    def __init__(self, in_channels):
        super(SpatialAttention, self).__init__()
        self.conv = nn.Conv2d(in_channels=in_channels,
out_channels=1, kernel_size=1)
        self.sigmoid = nn.Sigmoid()
    def forward(self, x):
        avg = torch.mean(x, dim=1, keepdim=True)
        max = torch.max(x, dim=1, keepdim=True)[0]
        out = torch.cat([avg, max], dim=1)
        out = self.conv(out)
        out = self.sigmoid(out)

        return out
```

Sample code 5.4: Spatial Attention implementation

5.6.2. Self Attention Implementation

```
class SelfAttention(nn.Module):
    def __init__(self, in_channels):
        super(SelfAttention, self).__init__()
            def forward(self, x): batch_size, C, H, W = x.size()

        proj_key = self.key_conv(x).view(batch_size, -1, H*W)
        energy = torch.bmm(proj_query, proj_key)
        attention = torch.softmax(energy, dim=-1)
        proj_value = self.value_conv(x).view(batch_size, -1, H*W)
        out = torch.bmm(proj_value, attention.permute(0, 2, 1))
        out = out.view(batch_size, C, H, W)
        out = self.gamma * out + x
        return out
```

Sample code 5.5: Self Attention implementation

5.6.3. Channel wise Attention Implementation

```
class ChannelWiseAttention(nn.Module):
    def __init__(self, in_channels, reduction=16):
        super(ChannelWiseAttention, self).__init__()

    def forward(self, x):
        avg_out = self.avg_pool(x)
        max_out = self.max_pool(x)
        out = avg_out + max_out
        out = self.fc(out)
        out = x * out
        return out
```

Sample code 5.6: Channel-wise Attention implementation

5.6.4. Cyclic Data Consistency Loss Implementation

Cyclic data consistency loss is used in MR image reconstruction to ensure that the output images produced by a model are consistent with the input images. In MR image reconstruction, the input images are usually under-sampled and contain artifacts, and the goal is to reconstruct a high-quality image from this under-sampled data.

By including this loss in the optimization process, our model is encouraged to produce an output image that is not only high-quality but also consistent with the input image. This helps to mitigate the impact of under-sampling and artifacts and improve the overall quality of the reconstructed MR image.

```
def cal_loss():
          #cyclic data consistency loss
  if cal_G:
        recon_img_AA = torch.mean((torch.abs((S01) - (S1))))
        error_img_AA = torch.mean(torch.abs((S01) - (Sp1)))
        smoothness_AA = torch.mean(total_variant(S1))
        recon_img_Aa = torch.mean(torch.abs((S01) - (T1)))
        error_img_Aa = torch.mean(torch.abs((S01) - (Tp1)))
        smoothness_Aa = torch.mean(total_variant(T1))
        ALPHA = 1e+1
        GAMMA = 1e-0
        DELTA = 1e-4
        RATES = torch.count_nonzero(torch.ones_like(mask)) / 2. /
torch.count_nonzero(mask)
        GAMMA = RATES
        cyc_loss = ALPHA * (recon_img_AA + error_img_AA) + GAMMA *
(recon_img_Aa + error_img_Bb)
```

Sample code 5.7: Cyclic Data Consistency Loss implementation

5.6.5. WGAN Loss Implementation

The build_loss function appears to compute the loss for a Wasserstein GAN (WGAN) given the discriminator's output for real and fake samples dis_real and dis_fake, respectively. Here's the PyTorch implementation:

```
def build_loss(dis_real, dis_fake):
    '''
    calculate WGAN loss
    '''
    d_loss = torch.mean(dis_fake - dis_real)
    g_loss = -torch.mean(dis_fake)
    return g_loss, d_loss
```

Sample code 5.8: WGAN loss implementation

In this implementation, d_loss is the difference between the mean of the discriminator's output for fake samples and the mean of its output for real samples. g_loss is the negative mean of the discriminator's output for fake samples. The generator tries to minimize g_loss while the discriminator tries to maximize it. The discriminator also tries to maximize d_loss. This formulation of the loss is intended to encourage the discriminator to output values that are as close as possible to the actual probability distribution of the real samples and to ensure that the gradient of the discriminator is well-defined almost everywhere.

5.6.6. Relative Average Discriminator Implementation

We used a relative average discriminator for MR image reconstruction in the discriminator part of GAN to improve the performance of the discriminator. Here is the implementation of RAD:

```
class RelativisticAvgDiscriminator(nn.Module):
    def forward(self, x):
        x_e0 = self.e0(x)
        x_e1 = self.e1(x_e0)
        x_e2 = self.e2(x_e1)
        x_e3 = self.e3(x_e2)
        ret = self.ret(x_e3)
 # apply average pooling to get the relative average discriminator
score
        avg_pool = nn.functional.avg_pool2d(ret,
ret.size()[2:]).view(ret.size()[0], -1)
        rel_avg_score = avg_pool / torch.norm(avg_pool, p=2,
dim=1, keepdim=True)
        return rel_avg_score
```

Sample code 5.9: Relative Average Discriminator implementation

Table 5.3: Content of the relative average discriminator architecture

Layers	Content of each block
1	Convolution - (filters = 64, kernel size=3, stride=2, padding=1, bias=False), Spectral normalization, Leaky ReLu
2	Convolution - (filters = 128, kernel size=3, stride=2, padding=1, bias=False), Spectral normalization, Leaky ReLu
3	Convolution - (filters = 256, kernel size=3, stride=2, padding=1, bias=False), Spectral normalization, Leaky ReLu
4	Convolution - (filters = 512, kernel size=3, stride=2, padding=1, bias=False), Spectral normalization, Leaky ReLu
5	Convolution - (filters = 512, kernel size= (4, 4), stride= (1, 1), padding= (2, 2))

5.7. Hyperparameter Configuration

A hyper-parameter refers to an established parameter that can be modified and has a direct impact on a model's training performance, such as batch size, number of epochs, optimization, learning rate, and loss of weight. The Adam optimizer was used in this study with a learning rate of 10^{-4} to update the network weights. Optimizers are techniques used to adjust the characteristics of a neural network, such as learning rate and weights, to reduce losses. Adam is known to combine the benefits of two stochastic gradient descent algorithms, namely AdaGrad and RMSProp (Kingma & Ba, 2015), and is simple to implement, computationally efficient, and memory-friendly. The step size, also known as the learning rate, determines the degree to which the weights are adjusted during training. The weight loss and learning rate used in this study are comparable to those reported in (G. Li et al., 2021). The model has trained over 15000 iterations, with a batch size of 4 and 300 numbers of epochs.

Table 5.3: Hyperparameter Configuration

Hyperparameter	**Values**
Batch size	4
Numbers of epochs	300
Number of iterations	15000
Optimizer	Adam optimizer
Learning Rate	0.001 ($10^{-4)}$

5.8. Experimental Class

It is required to conduct many experiments and test them to evaluate the essence of adopting hybrid attention mechanisms, relative average discriminator, and cyclic data consistency loss with WGAN loss in the improvement of reconstructed brain and knee MR image quality and its consistency.

Four (4) separate experiments for six classes of datasets are carried out as shown in table 5.4. In our works, totally we conducted twenty-four (24) experiment's. The total training times for all experiments are seventy-two (72) days. The implementation of the baseline work constitutes the first experiment. As stated in section 3.4, RSCA-GAN used dual residual U-net with spatial and channel-wise attention in its decoder parts of the generator network. The standard discriminator was used in its discriminator parts of the generative adversarial network. The loss function used in RSCA-GAN is cyclic data consistency loss.

The initial work of the proposed model (HARA-GAN) is the second experiment. In this experiment, we add spatial and self-attention mechanisms in the encoder parts of the residual U-net generator. In the third experiment, we conducted our experiment by replacing the standard discriminator with a relative average discriminator at the discriminator parts of GAN.

The overall proposed model is represented by the fourth experiment which included spatial and self-attention in encoder parts and spatial and channel-wise attention mechanisms in decoder parts of the residual U-net of the generator network. And also, the relative averages discriminator was implemented in discriminator networks with the combination of cyclic data consistency and WGAN loss for brain and knee MR image reconstruction with cartesian and non-cartesian under sampling rates of 12.5% and 10% respectively.

Table 5.4: List of experiment class

Notation	**Experiment**	**Dataset**
RSCA-GAN	Baseline	Calgary Campinas for Brain MR image and Stanford Fully Sampled 3D FSE for Knees MR image with 10% for two non-cartesian (radial and spiral) and 12.5% cartesian under sampling rate.
SSA + RSCA-GAN **(HARA-GAN-A)**	With the addition of spatial and self-attention at encoder parts of residual U-net	
SSA + RSCA-GAN + RAD **(HARA-GAN-B)**	By replacing the standard discriminator with the relative average at the discriminator parts of GAN	
SSA + RSCA-GAN + RAD + (CDC loss with WGAN loss) **(HARA-GAN-C)**	With the combination of cyclic data consistency loss and WGAN loss instead of using only cyclic data consistency loss	

CHAPTER SIX

6. RESULTS AND DISCUSSION

6.1. Chapter Overview

In this chapter, the outcomes of the preliminary research are examined, along with the various experiments that were conducted, including the proposed approach. The results are presented and compared using figures, graphs, and tables.

6.2. Experimental Results

We conducted training on multiple iterations of the proposed networks using various factors of under-sampling masks (10% for non-cartesian and 12.5% for cartesian) applied to the training datasets. To ensure robust results for this study, four experiments were performed, utilizing the same dataset, hardware, and software configuration for consistent comparison. To evaluate the effectiveness of our approach, we used two distinct datasets of magnetic resonance (MR) images sourced from the Calgary Campinas database (specifically, the brain MRI dataset) and the Stanford Fully Sampled 3D (the knee MRI dataset). By comparing our results with state-of-the-art CS-MRI reconstruction methods like RSCA-GAN, the proposed HARA-GAN shows the best quantitative values and the sharper edge of grey matter in brain and knee based on error map and quantitative evaluation metrics.

6.3. Results Based on Evaluation Metrics

The performance of the model was evaluated using three objective metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Normalized Mean Square Error (NMSE). The results of the evaluation are summarized in Table 6.1 and Table 6.2 for brain and knee MRI reconstruction respectively, which shows the reconstruction ability of the learned model by measuring the similarity between the restored images and the ground truth image (the Brain and knee MR image in this case). The evaluation was conducted using a dataset of 1700 slices for test brain MR images from the Calgary Campinas dataset, and 200 slices for test knee MR images from Mridata. Higher values of PSNR and SSIM, and lower values of NMSE indicate that the generated images are closer to the original images by their quality and consistency.

6.3.1. Evaluation Metrics for Brain MRI

Table 6.1: Quantitative results of Brain tests MRI

Mask	SR	Metrics	ZF	RSCA-GAN	HARA-GAN-A	HARA-GAN-B	HARA-GAN-C
Cartesian	12.5%	PSNR	23.09	30.49	30.50	31.38	**32.60**
		SSIM	0.69	0.86	0.87	0.88	**0.89**
		NMSE	12.13	3.40	3.02	2.53	**2.27**
Radial	10%	PSNR	23.14	30.19	30.71	31.17	**31.96**
		SSIM	0.52	0.84	0.84	0.84	**0.85**
		NMSE	11.61	2.76	2.75	2.61	**2.53**
Spiral	10%	PSNR	22.96	32.09	33.21	33.91	**34.63**
		SSIM	0.61	0.89	0.89	0.90	**0.91**
		NMSE	12.16	1.50	1.41	1.38	**1.34**

The bold value shows the best results in the experiments.

6.3.2. Evaluation Metrics for Knee MRI

Table 6.2: Quantitative results of Knee tests MRI

Mask	SR	Metrics	ZF	RSCA-GAN	HARA-GAN-A	HARA-GAN-B	HARA-GAN-C
Cartesian	12.5%	PSNR	28.11	32.08	33.61	34.55	**35.89**
		SSIM	0.74	0.83	0.85	0.85	**0.86**
		NMSE	8.10	3.25	2.50	2.29	**2.11**
Radial	10%	PSNR	27.95	32.82	33.66	34.75	**35.35**
		SSIM	0.68	0.82	0.86	0.87	**0.91**
		NMSE	8.36	3.17	2.89	2.55	**2.38**
Spiral	10%	PSNR	27.84	35.14	36.47	37.19	**38.74**
		SSIM	0.73	0.86	0.87	0.87	**0.89**
		NMSE	8.55	1.81	1.24	1.11	**1.09**

The bold value shows the best results in the experiments.

6.3.3. Statistical Significance of the Results

The statistical significance of the works can be assessed using a statistical test. This test helps to determine whether the observed results are statistically significant or not. In our study, we conducted the rank sum test(Deng et al., 2004) to assess the statistical significance of the differences in the PSNR, SSIM, and NMSE values between our proposed approach and the previous works. The resulting p-values for each metric were computed to be less than the conventional significance level of 0.05. This finding provides strong scientific evidence supporting the superiority of our proposed approach in terms of PSNR, SSIM, and NMSE when compared to the previous works. The p-values result in Table 6.3, Table 6.4, and Table 6.5 indicates that there are statistically significant differences between our approach and the previous works in terms of PSNR, SSIM, and NMSE.

Table 6.3: P-values of PSNR results for both brain and knee MRI

Under-sampling mask	P-values for Brain	P-values for Knee
Cartesian	0.002	0.009
Radial	0.004	0.016
Spiral	0.015	0.006

Table 6.4: P-values of SSIM results for both brain and knee MRI

Under-sampling mask	P-values for Brain	P-values for Knee
Cartesian	0.009	0.023
Radial	0.006	0.006
Spiral	0.001	0.001

Table 6.5: P-values of NMSE results for both brain and knee MRI

Under-sampling mask	P-values for Brain	P-values for Knee
Cartesian	0.016	0.043
Radial	0.008	0.018
Spiral	0.023	0.048

6.4. Reconstructed Results of the MR Images with Error Maps

6.4.1. Brain Cartesian MRI Result

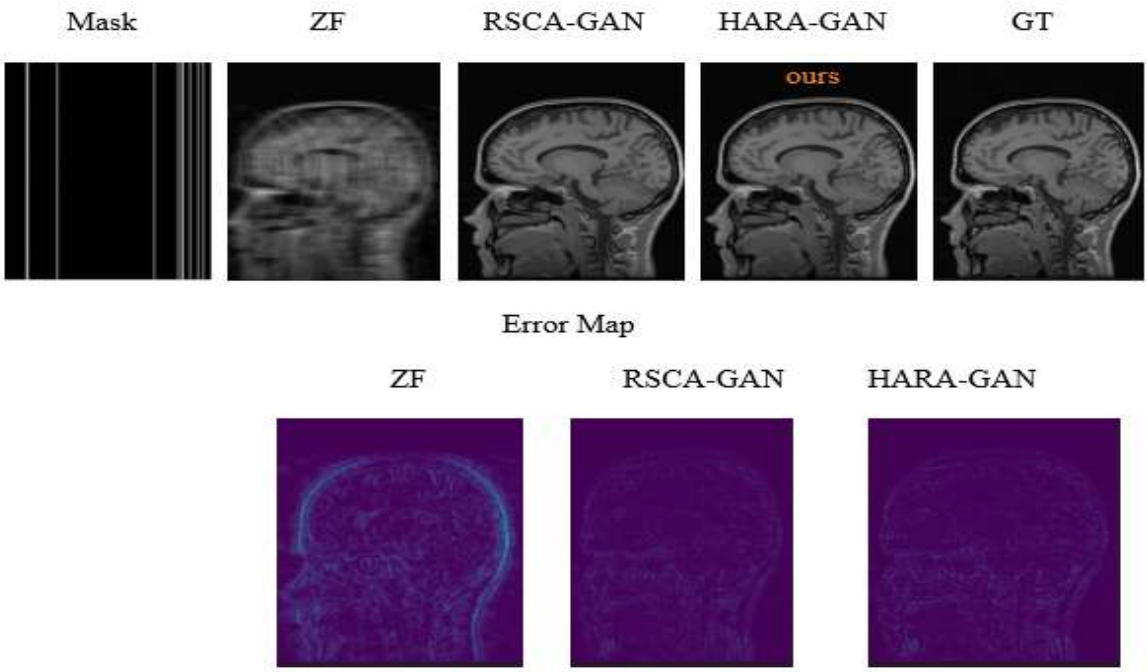


Figure 6.1: Brain cartesian under-sampled results

6.4.2. Knee Cartesian MRI Result

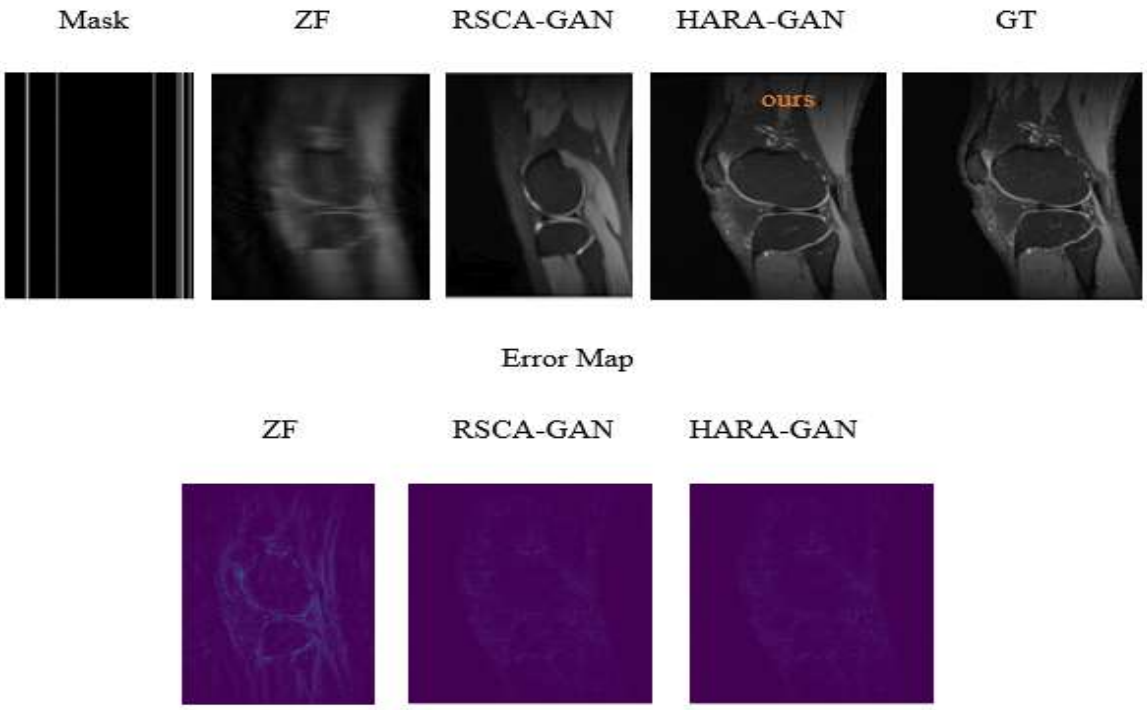


Figure 6.2: Knee cartesian under-sampled results

6.4.3. Brain Radial MRI Result

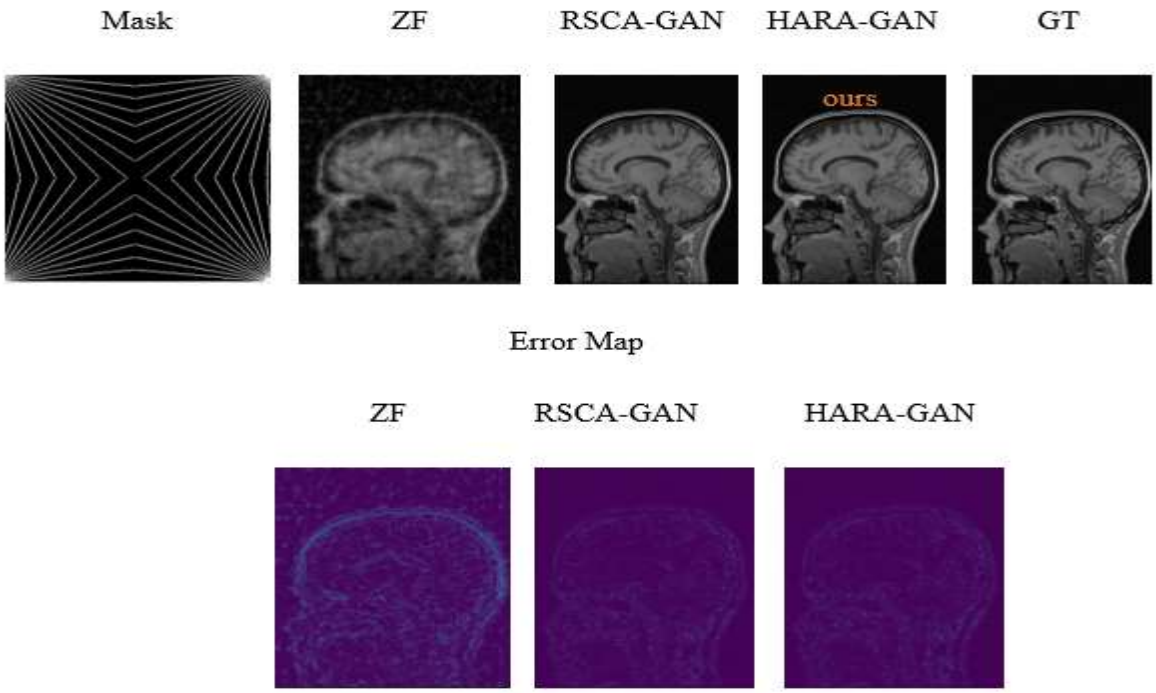


Figure 6.3: Brain Radial under-sampled results

6.4.4. Knee Radial MRI Result

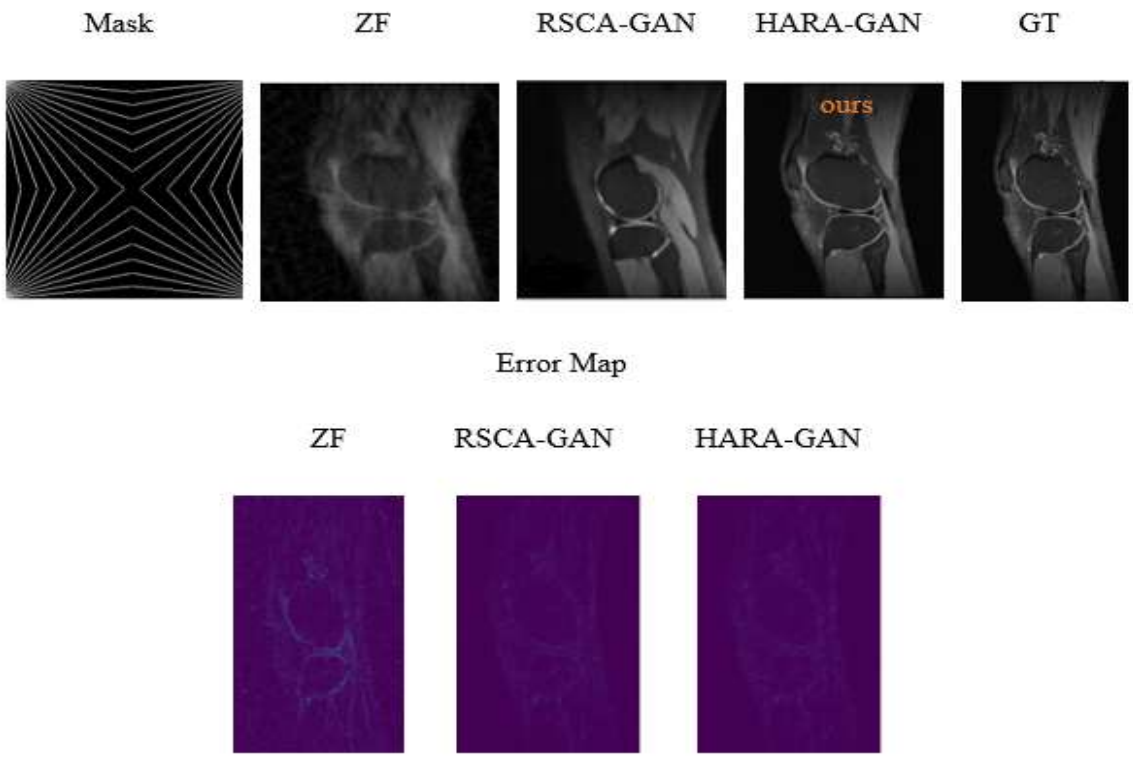


Figure 6.4: Knee radial under-sampled results

6.4.5. Brain Spiral MRI Result

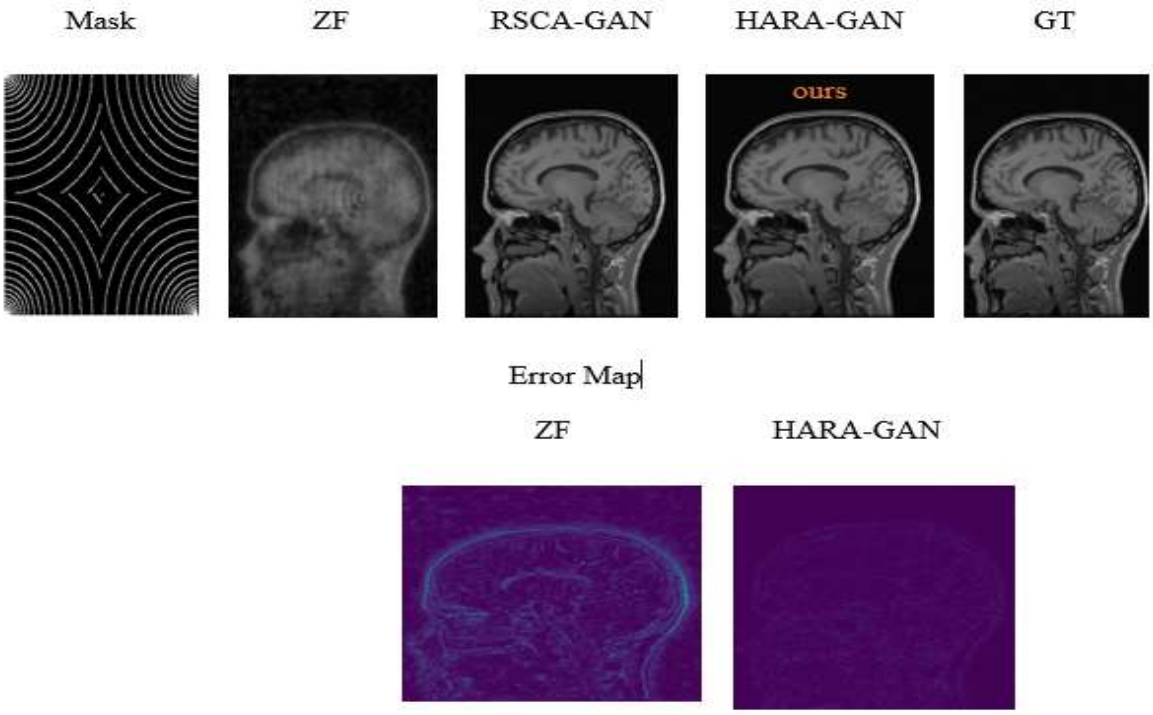


Figure 6.5: Brain spiral under-sampled results

6.4.6. Knee Spiral MRI Result

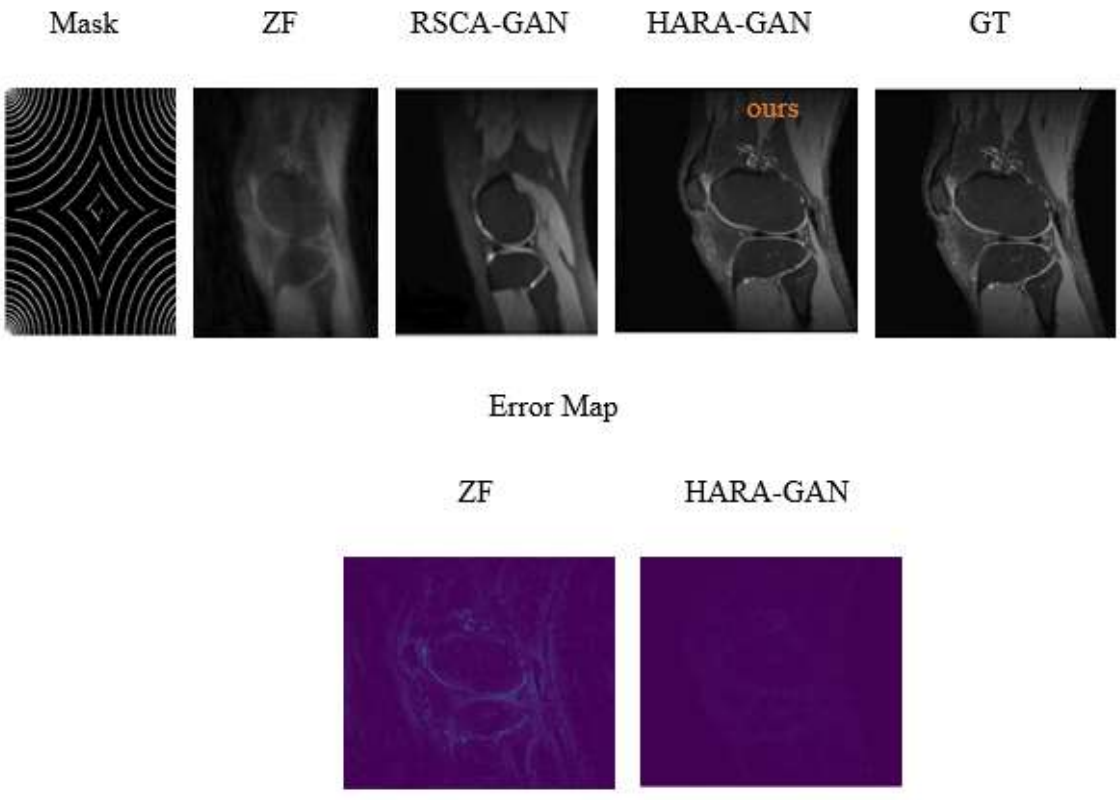


Figure 6.6: Knee spiral under-sampled results

6.5. Sample Training Log for HARA-GAN

During the training of our model, visual representations in the form of training log figures were used to assess and monitor the performance of the model. These log figures, including generator loss, discriminator loss, and reconstruction loss, among others, provide crucial insights into the training progress and effectiveness of the model for both brain and knee MR image reconstruction. By incorporating these visual elements, readers gain a comprehensive understanding of the model's performance, including convergence, component contributions, and areas for improvement. Analyzing the log figures reveals decreasing trends in generator and discriminator losses, confirming the effectiveness of our proposed approach and its ability to yield improved results. These log figures play a significant role in comprehending training dynamics and showcasing the evolving performance of our models, demonstrating the progress and optimization achieved in our research.

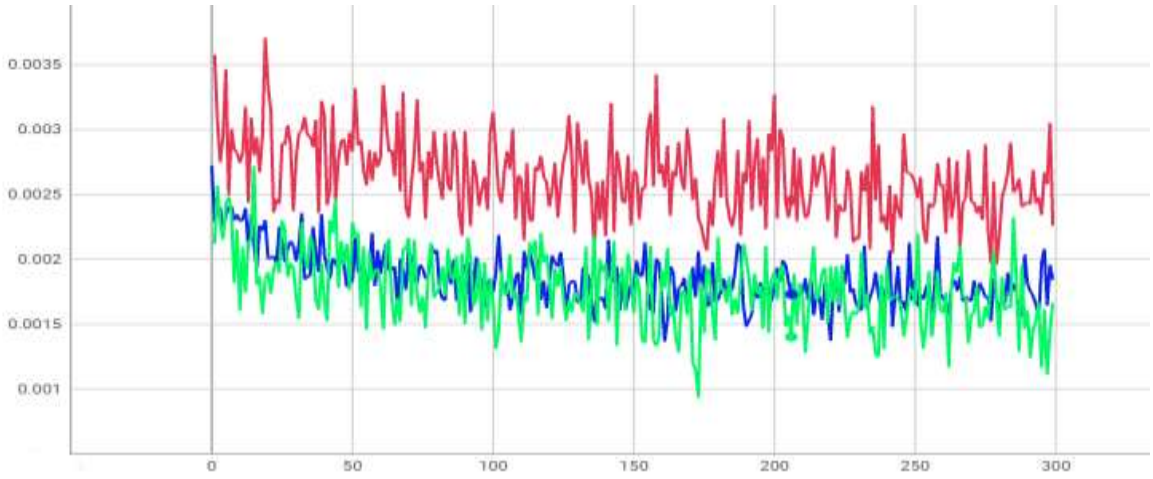


Figure 6.7: Reconstruction loss for brain MRI during training.

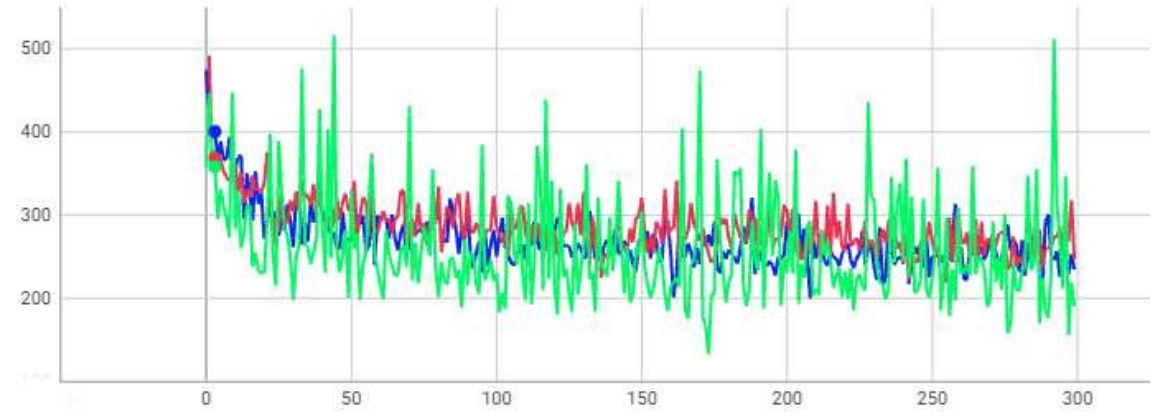


Figure 6.8: Total generator loss for brain MRI during training.

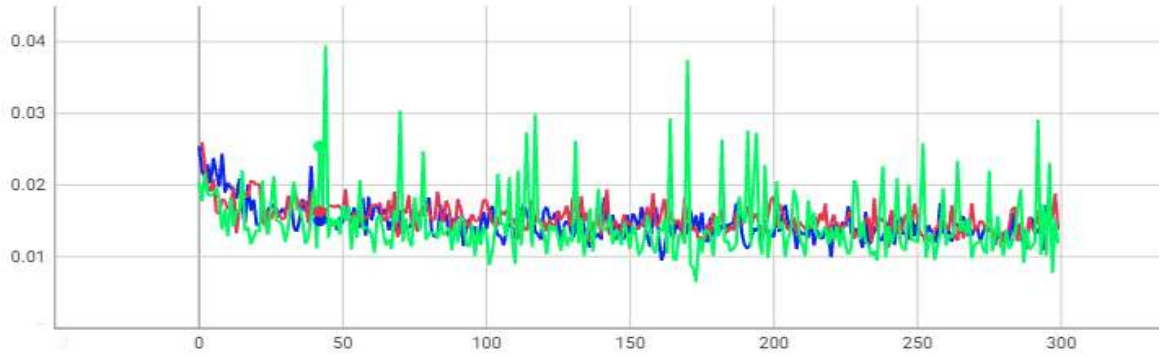


Figure 6.9: Discriminator loss for brain MRI during training.

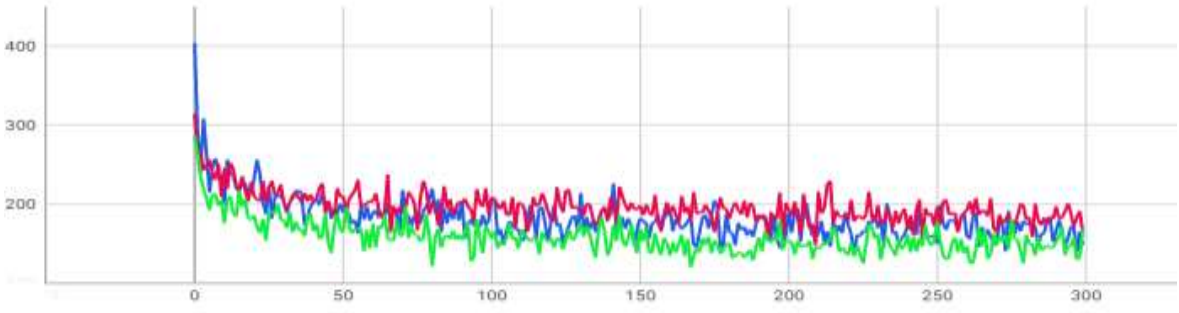


Figure 6.10: Total generator loss for knee MRI during training.

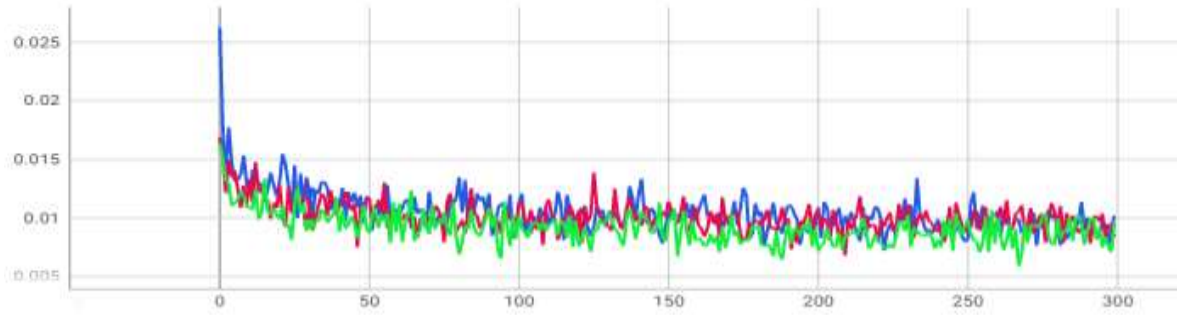


Figure 6.11: Reconstruction loss for knee MRI during training.

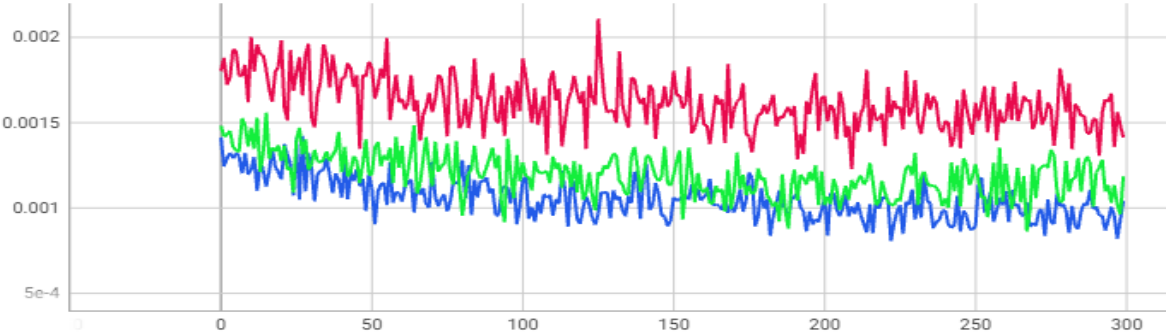


Figure 6.12: Discriminator loss for knee MRI during training.

- ✓ Red, green, and blue represents radial, spiral, and cartesian respectively.
- ✓ The X-axis indicates the loss scores and the Y-axis indicates the number of epochs.

6.6. Results Discussion

In this study, it was demonstrated that the proposed HARA-GAN is a viable method for reconstructing highly under-sampled MRI images using both Cartesian and non-Cartesian trajectories. HARA-GAN utilizes a combination of spatial self-attention, spatial channel-wise attention, residual U-net, and relative average discriminator. The results show that HARA-GAN performs better than other methods including DAGAN(G. Yang et al., 2018), RefineGAN(Quan et al., 2018), and RSCA-GAN(G. Li et al., 2021), according to various quantitative metrics. The SSA(R, 2021) shows the improved generator performance of the model, and the relative average discriminator(Jolicoeur-Martineau, 2019) improved the discriminator performance. In addition to that, the cyclic data consistency loss(Abu-Srhan et al., 2022) and WGAN loss (Arjovsky et al., 2017) were used to improve the MRI quality and its consistency compared with ground truth image. With the help of HARA-GAN, the image details, the edges, and details of information were reconstructed well.

6.6.1. Discussion on PSNR Results

The presented results in Figure 6.7 and Figure 6.8 shows the Peak Signal-to-Noise Ratio (PSNR) for three different reconstruction methods, HARA-GAN-A, HARA-GAN-B, and HARA-GAN-C, applied to under-sampled MRI brain and knee data. The comparison is made against two other methods, ZF which represents the under-sampled data, and RSCA-GAN which is the baseline method. The PSNR values are reported for three different sampling patterns: Cartesian, Radial, and Spiral.

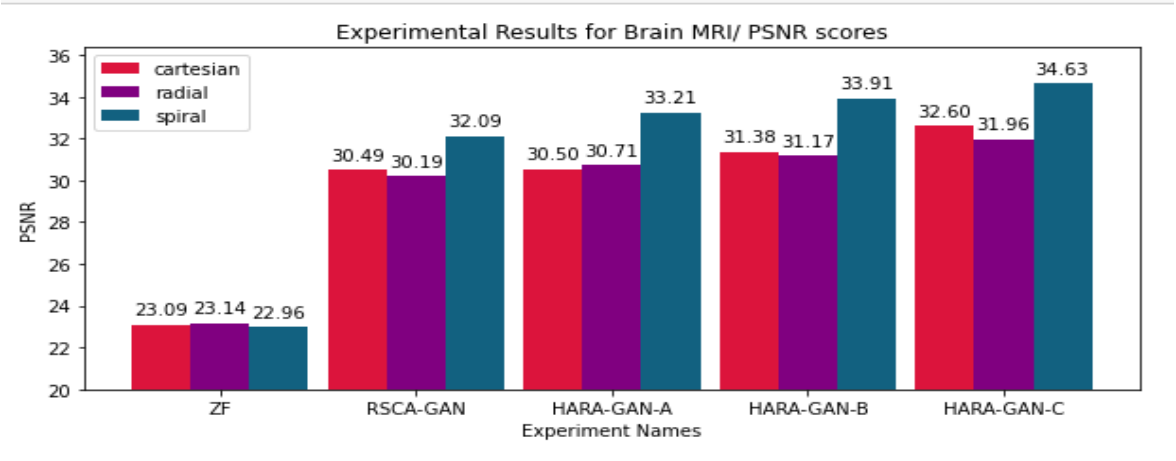


Figure 6.13: PSNR scores for brain MRI reconstruction

- ✓ The X-axis indicates the PSNR scores and the Y-axis indicates the experimental class.

As shown in Figure 6.7, the result indicates as HARA-GAN-C outperforms the baseline method for all sampling patterns by improving 30.49 to **32.60** for cartesian, 30.19 to **31.96** for radial, and 32.09 to **34.63** PSNR across all sampling patterns for brain MRI reconstruction.

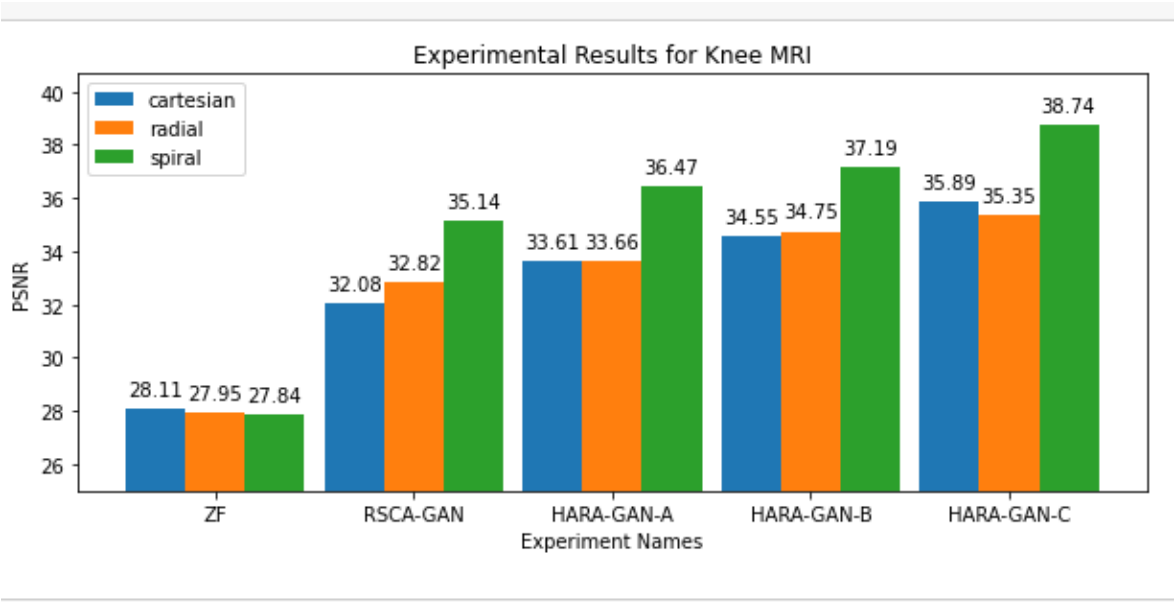


Figure 6.14: PSNR scores for knee MRI reconstruction

- ✓ The X-axis indicates the PSNR scores and the Y-axis indicates the experimental class.

As shown in Figure 6.8, the result indicates as HARA-GAN-C outperforms the baseline method for all sampling patterns by improving 32.08 to **35.89** for cartesian, 32.82 to **35.35** for radial, and 35.14 to **38.74** PSNR across all sampling patterns for knee MRI reconstruction. HARA-GAN-A and HARA-GAN-B also outperform the baseline method RSCA-GAN by achieving the highest PSNR across all sampling patterns for both brain and knee MR image reconstruction.

6.6.2. Discussion on SSIM Results

The SSIM (Structural Similarity Index) results presented in Figure 6.9 and Figure 6.10 shows the performance of three different brains and knee MRI reconstruction methods applied to under-sampled MRI data. The three reconstruction methods are HARA-GAN-A, HARA-GAN-B, and HARA-GAN-C, and they are compared against two other methods: ZF, which represents the under-sampled data, and RSCA-GAN, which is the baseline method. The SSIM values are reported for three different sampling patterns: Cartesian, Radial, and Spiral.

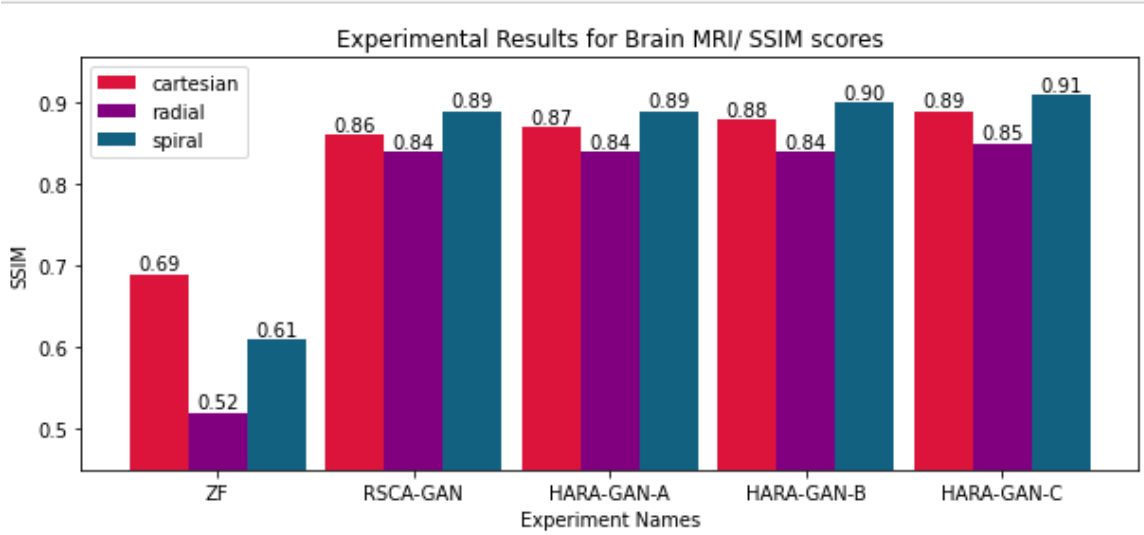


Figure 6.15: SSIM scores for brain MRI reconstruction

- ✓ The X-axis indicates the SSIM scores and the Y-axis indicates the experimental class.

The result in Figure 6.9 indicates as HARA-GAN-C outperforms all other methods across all sampling patterns, achieving an SSIM value from RSCA-GAN results of 0.86 to **0.89** for cartesian, 0.84 to **0.85** for radial, 0.89 to **0.91** for spiral sampling which is the highest value reported in this study for brain MRI reconstruction.

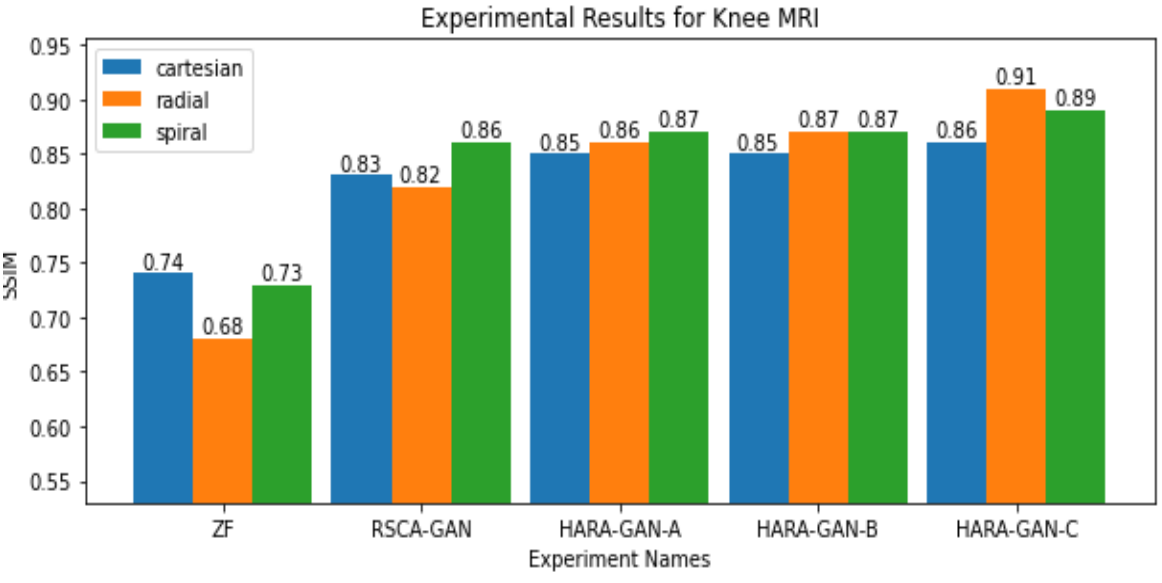


Figure 6.16: SSIM scores for knee MRI reconstruction

- ✓ The X-axis indicates the SSIM scores and the Y-axis indicates the experimental class.

As shown in Figure 6.10, HARA-GAN-C outperforms all other methods across all sampling patterns, achieving an SSIM value from RSCA-GAN results of 0.83 to **0.86** for cartesian, 0.82 to **0.91** for radial, 0.86 to **0.89** for spiral sampling which is the highest value reported in this study for knee MRI reconstruction.

Compared to the baseline work RSCA-GAN, all three HARA-GAN methods show improvements in SSIM, with HARA-GAN-C achieving the largest improvement. ZF, which represents the under-sampled data, performs the worst across all sampling patterns, which is expected since it does not contain complete information.

6.6.3. Discussion on NMSE Results

In this study, the NMSE metric was used alongside PSNR and SSIM to assess the quality of the reconstructed MRI images. The comparison is made against two other methods, ZF which represents the under-sampled data, and RSCA-GAN which is the baseline work. The NMSE values are reported for three different sampling patterns: Cartesian, Radial, and Spiral.

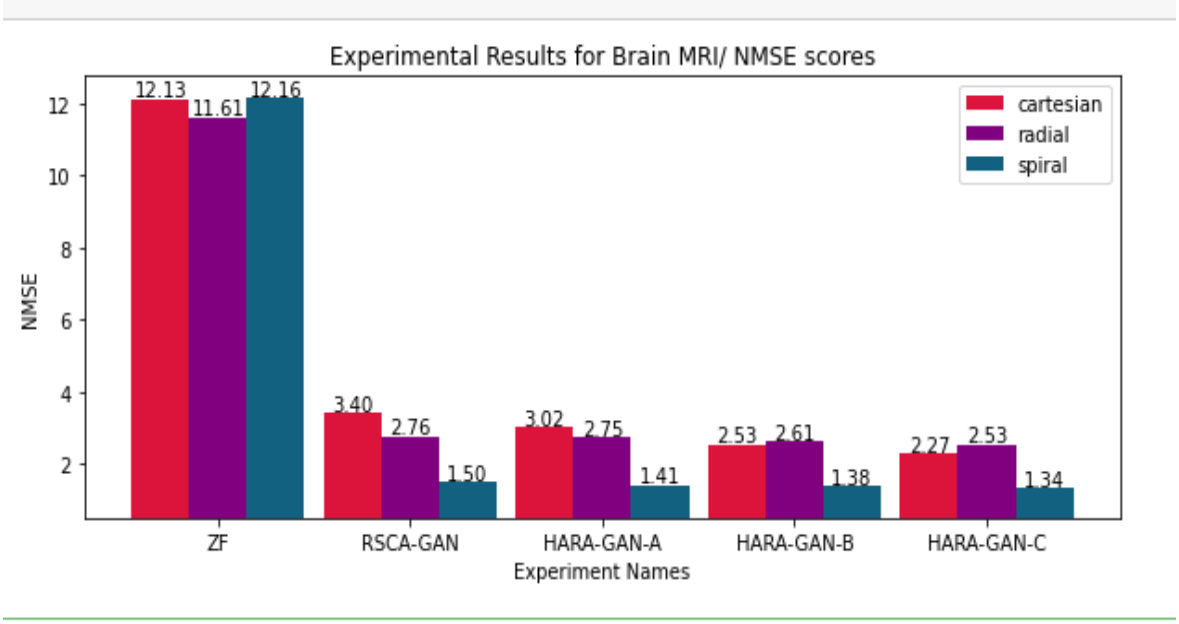


Figure 6.17: NMSE scores for Brain MRI reconstruction

- ✓ The X-axis indicates the NMSE scores and the Y-axis indicates the experimental class.

Figure 6.11 shows that RSCA-GAN has the highest NMSE values for brain MRI reconstruction, with scores of 3.40 for cartesian, 2.76 for radial, and 1.50 for spiral sampling. In contrast, HARA-GAN-C has lower NMSE values, with scores of **2.27** for cartesian, **2.53**

for radial, and **1.34** for spiral sampling, indicating that the quality of the generated brain MRI image is better than in previous experiments.

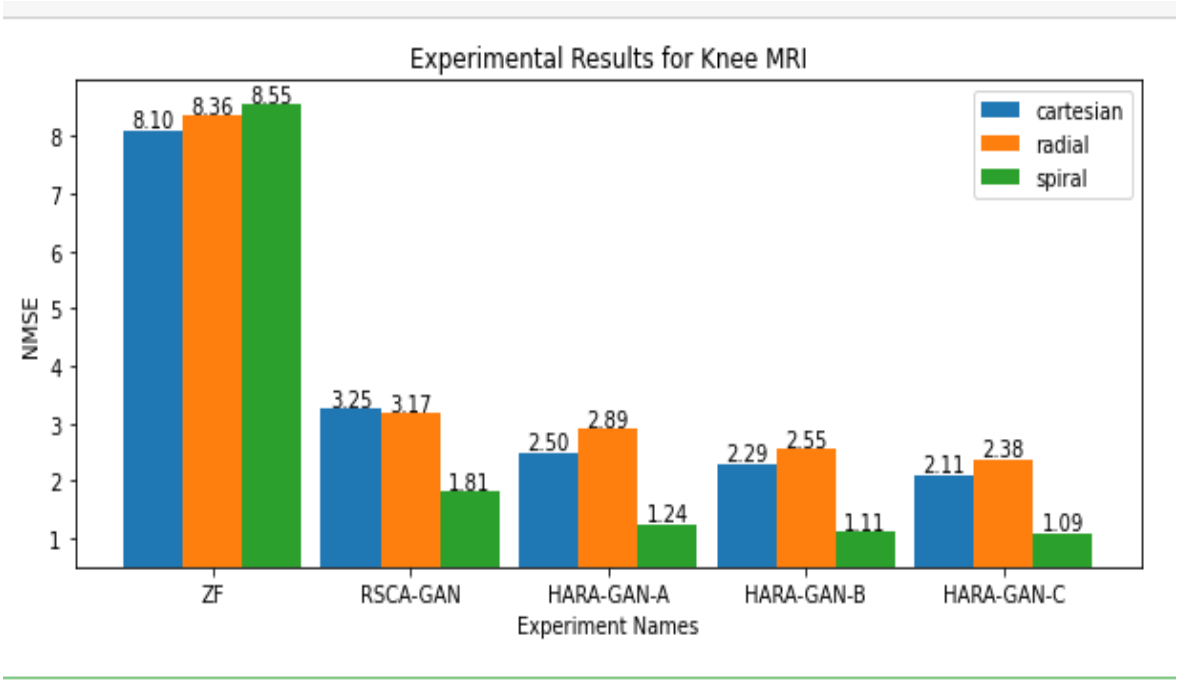


Figure 6.18: NMSE scores for knee MRI reconstruction

- ✓ The X-axis indicates the NMSE scores and the Y-axis indicates the experimental class.

And also, Figure 6.12 shows that RSCA-GAN has the highest NMSE values for knee MRI reconstruction, with scores of 3.25 for cartesian, 3.17 for radial, and 1.81 for spiral sampling. In contrast, HARA-GAN-C has lower NMSE values, with scores of **2.11** for cartesian, **2.38** for radial, and **1.09** for spiral sampling, indicating that the quality of the generated knee MRI image is better than in previous experiments. HARA-GAN-A and HARA-GAN-B achieve better results than RSCA-GAN for both brain and knee MRI reconstruction, but they achieve higher NMSE values than HARA-GAN-C. HARA-GAN-C has the lowest scores among all experiments, which suggests that the model generates image sequences of higher quality than the previous experiments.

The results indicate that HARA-GAN-C achieves high performance across all sampling patterns and all experiments, which suggests that the model generates image sequences of higher quality than the previous experiments. HARA-GAN-A and HARA-GAN-B also outperform the baseline method RSCA-GAN for all sampling patterns.

6.6.4. Research Question and Answer Discussion

This study provided answers to all the research questions "**1.** How hybrid attention mechanism and relative Average discriminator is implemented with GAN to improve MR image reconstruction?, **2.** What are the effects of the hybrid Attention mechanism for reconstructing low under-sampled MR images?, **3.** What are the effects of relative average discriminator in standard GAN for MR image reconstruction?, **4.** What effect would Wasserstein GAN loss with cyclic data consistency loss have on the quality and consistency of MR image reconstruction?". The subsequent section outlines how each question is addressed.

Answer for Q1: As stated in **section 4.2**, the study found certain learning techniques for enhancing the reconstructed MR image quality and consistency through dual residual U-net at generator parts and relative average discriminator at discriminator parts of GAN. The residual U-net was implemented using spatial self-attention at the encoder module, and a spatial channel-wise attention mechanism at the decoder module was identified for improving the structural details of the reconstructed MR image.

Answer for Q2: The study incorporated a spatial self-attention mechanism into the generator network as a useful residual U-net block. The effectiveness of the spatial self-attention mechanism in improving the quality of the reconstructed low under-sampled MR image was demonstrated through the PSNR, SSIM, and NMSE results based on the HARA-GAN-A experiment in **table 6.1** for brain and **table 6.2** for knee MR images.

Answer for Q3: The relative average discriminator shows the effectiveness of the network for both knee and brain MR image reconstruction with cartesian and non-cartesian trajectories (radial and spiral) sampling. It solves the mode collapse and instability problems in standard GAN that degrades the quality of GAN to reconstruct high-quality MRI from low under-sampled data. The relative average discriminator based on the HARA-GAN-B experiment, shows an improvement in MR image quality based on PSNR, SSIM, and NMSE scores in **Table 6.1** for the brain and in **Table 6.2** for the knee.

Answer for Q4: WGAN loss with cyclic data consistency loss shows the effectiveness of the networks for brain and knee MR image reconstruction. Based on the evaluation metrics as shown in **table 6.1** and **table 6.2**, HARA-GAN-C experimental results improves PSNR, SSIM, and NMSE scores for both brain and knee with all sampling patterns for MRI reconstruction.

CHAPTER SEVEN

7. CONCLUSION AND FUTURE WORKS

7.1. Conclusion

In this article, we presented a new approach for solving the MR image reconstruction problem using a deep learning-based generative adversarial network model called HARA-GAN. It focused on low under-sampled rates for both brain and knee MRI data reconstruction. For cartesian 12.5% (8x faster scan) and for non-cartesian (radial and spiral) 10% (10x faster scan) of the k-space data point were used to reconstruct under-sampled MR images that related to fully sampled MRI data. Generating high-quality MR images from low under-sampled input MR images is a challenging research problem. Compressed sensing MRI can reduce scan time and cost for clinical applications, but reconstruction quality is often compromised. Previous approaches, such as RSCA-GAN, have limitations in low under-sampled rates and fail to improve image quality or address missing concepts in the discriminator's input data. In general, this study presented a novel approach for MR image reconstruction using a Generative Adversarial Network (GAN) architecture. The architecture, inspired by generative adversarial networks with Residual U-net and Hybrid attention mechanism is designed with a deeper generator G network and trained adversarially with the relative average discriminator D, along with cyclic data consistency and WGAN loss to achieve the better reconstruction of the very low under-sampled k-space data for both brain and knee MR image reconstruction.

The objective of this research was to improve the performance of generative adversarial networks (GANs) for magnetic resonance imaging (MRI) reconstruction. To achieve this, we proposed an improved GAN called HARA-GAN, which combines spatial self-attention in encoder parts of residual U-net and spatial channel-wise attention in decoder parts of residual U-net. This method can effectively reconstruct MR images even under high acceleration factors that mean a very low under-sampled rate, using both Cartesian and non-Cartesian under-sampling masks.

The proposed model, named HARA-GAN, employs spatial self-attention in the encoder and spatial channel-wise attention in the Residual U-Net, while using a relative average discriminator at discrimination. The use of spatial self-attention and spatial channel-wise attention contributed to the improved performance of the model, which highlights the importance of these attention mechanisms in the GAN architecture. The study also

discovered cyclic data consistency loss with WGAN loss to compute the edge and details loss between the reconstructed MR image and fully sampled MR images, which enables the reconstruction network to enhance the specific features needed by the discriminator network. The loss function used in the training process was cyclic data consistency with WGAN loss.

Several experiments were conducted simultaneously to make comparisons. The first experiment is HARA-GAN-A which used a spatial self-attention mechanism in encoder parts of the residual U-net that improve the generator performance compared to RSCA-GAN.

The second experiment HARA-GAN-B was conducted using a relative average discriminator in place of a standard discriminator. It shows the effectiveness relative average discriminator compared to the standard discriminator used in RSCA-GAN by improving discriminator performance for HARA-GAN. The third experiment HARA-GAN-C also conducted using cyclic data consistency loss with WGAN loss. HARA-GAN-C is a promising method for reconstructing under-sampled brain and knee MRI data and the most effective in reconstructing high-quality and consistent brain and knee MR images among all the experiments based on PSNR, SSIM, and NMSE scores.

The experimental results demonstrate that HARA-GAN outperforms existing MR image reconstruction methods based on various quantitative metrics in terms of both image quality and consistency. Overall, the results demonstrate the effectiveness of the HARA-GAN methods for knee and brain MRI reconstruction. These findings have important implications for clinical applications, as high-quality and consistent MRI images are crucial for accurate diagnosis and treatment of brain and knee problems. Therefore, we believe that the application of HARA-GAN has great potential in clinical practice.

So, this proposed “Hybrid Attention and Relative Average Discriminator based on GANs for MR Image Reconstruction” has tried to contribute further improvement on the generative adversarial network architecture to produce better results that have high quality and consistency to compared the fully sampled MR images for both brain and knee under-sampled MRI data. Further, this study has contributed technical level improvements with the addition of different mechanisms and constraints to improve the qualities and performance of the work.

7.2. Future Works

Our future work involves conducting a more in-depth analysis of HARA-GAN to better understand its architecture and exploring the use of incredibly deep multi-fold chains to improve reconstruction accuracy based on its target-driven characteristic. Additionally, we put as future works to extend HARA-GAN's capabilities to handle dynamic MRI and to develop a distributed version for parallel training and deployment on a cluster system.

However, our study is subject to some limitations. One such limitation is that the input of our network is a single-channel image, whereas clinical scanners typically produce multi-channel images. Therefore, it is necessary to synthesize the multi-channel MR image into a single-channel format, which requires an additional workload.

In future works, we plan to make two major changes to our approach. Firstly, we aim to modify the input of the network from single-channel to multi-channel, which aligns with the requirements of clinical routine. Additionally, our current network utilizes supervised learning during the training process. However, when it is challenging to obtain fully-sampled data, we will use an unsupervised training method to train the network.

Future work could involve extending the experiments to include MR image reconstruction of additional body parts beyond the brain and knee, to evaluate the effectiveness of the proposed method for a wider range of clinical applications. Additionally, researchers may explore ways to further improve the performance of the proposed method by incorporating prior information of the discriminator's input data or by using other advanced deep learning techniques.

* * *

Special Acknowledgment

This research work was funded by grant number ASTU/SM-R/811/23 from Adama Science and Technology University, **Adama, Ethiopia**.

REFERENCES

Abu-Srhan, A., Abushariah, M. A. M., & Al-Kadi, O. S. (2022). The effect of loss function on conditional generative adversarial networks. *Journal of King Saud University - Computer and Information Sciences*, *34*(9), 6977–6988. https://doi.org/10.1016/j.jksuci.2022.02.018

Ahishakiye, E., Van Gijzen, M. B., Tumwiine, J., Wario, R., & Obungoloch, J. (2021). A survey on deep learning in medical image reconstruction. *Intelligent Medicine*, *1*(3), 118–127. https://doi.org/10.1016/J.IMED.2021.03.003

Arjovsky, M., Chintala, S., & Bottou, L. (2017). *Wasserstein GAN*. http://arxiv.org/abs/1701.07875

Boyd, S., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. (2011). Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends® in Machine Learning*, *3*(1), 1–122. https://doi.org/10.1561/2200000016

Chen, T., Song, X., & Wang, C. (2018). Preserving-Texture Generative Adversarial Networks for Fast Multi-Weighted MRI. *IEEE Access*, *6*, 71048–71059. https://doi.org/10.1109/ACCESS.2018.2877932

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. (2018). Generative Adversarial Networks: An Overview. *IEEE Signal Processing Magazine*, *35*(1), 53–65. https://doi.org/10.1109/MSP.2017.2765202

Deng, L., Pei, J., Ma, J., & Lee, D. L. (2004). *Industry / Government Track Paper A Rank Sum Test Method for Informative Gene Discovery ∗*. 410–419.

Deshmane, A., Gulani, V., Griswold, M. A., & Seiberlich, N. (2012). Parallel MR imaging. *Journal of Magnetic Resonance Imaging : JMRI*, *36*(1), 55–72. https://doi.org/10.1002/jmri.23639

Emami, H., Aliabadi, M. M., Dong, M., & Chinnam, R. B. (2021). SPA-GAN: Spatial Attention GAN for Image-to-Image Translation. *IEEE Transactions on Multimedia*, *23*, 391–401. https://doi.org/10.1109/TMM.2020.2975961

Eo, T., Jun, Y., Kim, T., Jang, J., Lee, H.-J., & Hwang, D. (2018). KIKI-net: cross-domain convolutional neural networks for reconstructing undersampled magnetic resonance

images. *Magnetic Resonance in Medicine*, *80*(5), 2188–2201.

Esteva, A., Chou, K., Yeung, S., Naik, N., Madani, A., Mottaghi, A., Liu, Y., Topol, E., Dean, J., & Socher, R. (2021). Deep learning-enabled medical computer vision. *Npj Digital Medicine*, *4*(1), 5. https://doi.org/10.1038/s41746-020-00376-2

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, *3*(January), 2672–2680. https://doi.org/10.3156/jsoft.29.5_177_2

Griswold, M. A., Jakob, P. M., Heidemann, R. M., Nittka, M., Jellus, V., Wang, J., Kiefer, B., & Haase, A. (2002). Generalized Autocalibrating Partially Parallel Acquisitions (GRAPPA). *Magnetic Resonance in Medicine*, *47*(6), 1202–1210. https://doi.org/10.1002/mrm.10171

Hamilton, J., Franson, D., & Seiberlich, N. (2017). Recent advances in parallel imaging for MRI. *Progress in Nuclear Magnetic Resonance Spectroscopy*, *101*, 71–95.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-December*, 770–778. https://doi.org/10.1109/CVPR.2016.90

Huang, J., Wu, Y., Wu, H., & Yang, G. (2022). Fast MRI Reconstruction: How Powerful Transformers Are? *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, *2022-July*, 2066–2070. https://doi.org/10.1109/EMBC48229.2022.9871475

Huang, J., Zhang, S., & Metaxas, D. (2010). Efficient MR image reconstruction for compressed MR imaging. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *6361 LNCS*(PART 1), 135–142. https://doi.org/10.1007/978-3-642-15705-9_17/COVER

Huang, Q., Yang, D., Wu, P., Qu, H., Yi, J., & Metaxas, D. (2019). MRI reconstruction via cascaded channel-wise attention network. *Proceedings - International Symposium on Biomedical Imaging*, *2019-April*, 1622–1626. https://doi.org/10.1109/ISBI.2019.8759423

Jaspan, O. N., Fleysher, R., & Lipton, M. L. (2015). Compressed sensing MRI: A review of

the clinical literature. *British Journal of Radiology*, *88*(1056), 1–12. https://doi.org/10.1259/bjr.20150487

Jay, F., Renou, J.-P., Voinnet, O., & Navarro, L. (2017). Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks Jun-Yan. *Proceedings of the IEEE International Conference on Computer Vision*, 183–202. http://link.springer.com/10.1007/978-1-60327-005-2_13

Jiang, W., Li, Q., Zhan, K., Fang, Y., & Shen, F. (2022). Hybrid attention network for image captioning. *Displays*, 102238.

Jolicoeur-Martineau, A. (2019). The relativistic discriminator: A key element missing from standard GaN. *7th International Conference on Learning Representations, ICLR 2019*.

Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–15.

Knyaz, V. A., Kniaz, V. V., & Remondino, F. (2019). Image-to-voxel model translation with conditional adversarial networks. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11129 LNCS*, 601–618. https://doi.org/10.1007/978-3-030-11009-3_37

Kojima, S., Shinohara, H., Hashimoto, T., & Suzuki, S. (2018). Undersampling patterns in k-space for compressed sensing MRI using two-dimensional Cartesian sampling. *Radiological Physics and Technology*, *11*(3), 303–319. https://doi.org/10.1007/s12194-018-0469-y

Lazarus, C., Weiss, P., Chauffert, N., Mauconduit, F., El Gueddari, L., Destrieux, C., Zemmoura, I., Vignaud, A., & Ciuciu, P. (2019). SPARKLING: variable-density k-space filling curves for accelerated T2*-weighted MRI. *Magnetic Resonance in Medicine*, *81*(6), 3643–3661. https://doi.org/10.1002/mrm.27678

Li, G., Lv, J., & Wang, C. (2021). A Modified Generative Adversarial Network Using Spatial and Channel-Wise Attention for CS-MRI Reconstruction. *IEEE Access*, *9*, 83185–83198. https://doi.org/10.1109/ACCESS.2021.3086839

Li, X., Ding, L., Wang, L., & Cao, F. (2018). FPGA accelerates deep residual learning for image recognition. *Proceedings of the 2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2017*, *2018-*

January, 837–840. https://doi.org/10.1109/ITNEC.2017.8284852

Li, Y., Li, J., Ma, F., Du, S., & Liu, Y. (2021). High quality and fast compressed sensing MRI reconstruction via edge-enhanced dual discriminator generative adversarial network. *Magnetic Resonance Imaging*, *77*, 124–136.

Lustig, M., Donoho, D. L., Santos, J. M., & Pauly, J. M. (2008). Compressed sensing MRI: A look at how CS can improve on current imaging techniques. *IEEE Signal Processing Magazine*, *25*(2), 72–82. https://doi.org/10.1109/MSP.2007.914728

Lustig, M., Santos, J. M., Lee, J.-H., Donoho, D. L., & Pauly, J. M. (2005). Application of compressed sensing for rapid MR imaging. *SPARS,(Rennes, France)*.

Mardani, M., Gong, E., Cheng, J. Y., Vasanawala, S. S., Zaharchuk, G., Xing, L., & Pauly, J. M. (2019). Deep generative adversarial neural networks for compressive sensing MRI. *IEEE Transactions on Medical Imaging*, *38*(1), 167–179. https://doi.org/10.1109/TMI.2018.2858752

Mirza, M., & Osindero, S. (2014). *Conditional Generative Adversarial Nets*. 1–7. http://arxiv.org/abs/1411.1784

Osman AM, V. S. (2018). Self-Attention Generative Adversarial Networks. *ICTAS Conference on Information Communications Technology and Society.*

Pang, Y., Lin, J., Qin, T., & Chen, Z. (2022). Image-to-Image Translation: Methods and Applications. *IEEE Transactions on Multimedia*, *24*, 3859–3881. https://doi.org/10.1109/TMM.2021.3109419

Paschal, C. B., & Morris, H. D. (2004). K-Space in the Clinic. *Journal of Magnetic Resonance Imaging*, *19*(2), 145–159. https://doi.org/10.1002/jmri.10451

Quan, T. M., Nguyen-Duc, T., & Jeong, W. K. (2018). Compressed Sensing MRI Reconstruction Using a Generative Adversarial Network With a Cyclic Loss. *IEEE Transactions on Medical Imaging*, *37*(6), 1488–1497. https://doi.org/10.1109/TMI.2018.2820120

R, J. V. C. I. (2021). *Spatial self-attention network with self-attention distillation for fine-grained image recognition ☆. 81*.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Lecture Notes in Computer Science (Including*

Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), *9351*, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28/COVER

Sara, U., Akter, M., & Uddin, M. S. (2019). Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study. *Journal of Computer and Communications*, *07*(03), 8–18. https://doi.org/10.4236/jcc.2019.73002

Sawyer, A. M., Lustig, M., Alley, M., Uecker, P., Virtue, P., Lai, P., & Vasanawala, S. (2013). *Creation of fully sampled MR data repository for compressed sensing of the knee*.

Schlemper, J., Caballero, J., Hajnal, J. V., Price, A., & Rueckert, D. (2017). A deep cascade of convolutional neural networks for MR image reconstruction. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10265 LNCS*, 647–658. https://doi.org/10.1007/978-3-319-59050-9_51/COVER

Shende, P., Pawar, M., & Kakde, S. (2019). A brief review on: MRI images reconstruction using GAN. *Proceedings of the 2019 IEEE International Conference on Communication and Signal Processing, ICCSP 2019*, 139–142. https://doi.org/10.1109/ICCSP.2019.8698083

Souza, R., Lucena, O., Garrafa, J., Gobbi, D., Saluzzi, M., Appenzeller, S., Rittner, L., Frayne, R., & Lotufo, R. (2018). An open, multi-vendor, multi-field-strength brain MR dataset and analysis of publicly available skull stripping methods agreement. *NeuroImage*, *170*, 482–494. https://doi.org/10.1016/J.NEUROIMAGE.2017.08.021

Sun, L., Fan, Z., Ding, X., Huang, Y., & Paisley, J. (2019). Joint CS-MRI Reconstruction and Segmentation with a Unified Deep Network. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11492 LNCS*, 492–504. https://doi.org/10.1007/978-3-030-20351-1_38/COVER

Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018). *CBAM: Convolutional Block Attention Module* (pp. 3–19).

yang, yan, Sun, J., Li, H., & Xu, Z. (2016). Deep ADMM-Net for Compressive Sensing MRI. *Advances in Neural Information Processing Systems*, *29*.

Yang, G., Yu, S., Dong, H., Slabaugh, G., Dragotti, P. L., Ye, X., Liu, F., Arridge, S.,

Keegan, J., Guo, Y., & Firmin, D. (2018). DAGAN: Deep De-Aliasing Generative Adversarial Networks for Fast Compressed Sensing MRI Reconstruction. *IEEE Transactions on Medical Imaging*, *37*(6), 1310–1321. https://doi.org/10.1109/TMI.2017.2785879

Yang, X. (2020). An Overview of the Attention Mechanisms in Computer Vision. *Journal of Physics: Conference Series*, *1693*(1), 012173. https://doi.org/10.1088/1742-6596/1693/1/012173

Ye, J. C. (2019). Compressed sensing MRI: a review from signal processing perspective. *BMC Biomedical Engineering*, *1*(1), 1–17. https://doi.org/10.1186/s42490-019-0006-z

Yin, X.-X., Hadjiloucas, S., & Zhang, Y. (2017). *Introduction and Motivation for Conducting Medical Image Analysis*. 1–26. https://doi.org/10.1007/978-3-319-57027-3_1

Yuan, Z., Jiang, M., Wang, Y., Wei, B., Li, Y., Wang, P., Menpes-Smith, W., Niu, Z., & Yang, G. (2020). SARA-GAN: Self-Attention and Relative Average Discriminator Based Generative Adversarial Networks for Fast Compressed Sensing MRI Reconstruction. *Frontiers in Neuroinformatics*, *14*, 58. https://doi.org/10.3389/FNINF.2020.611666/BIBTEX

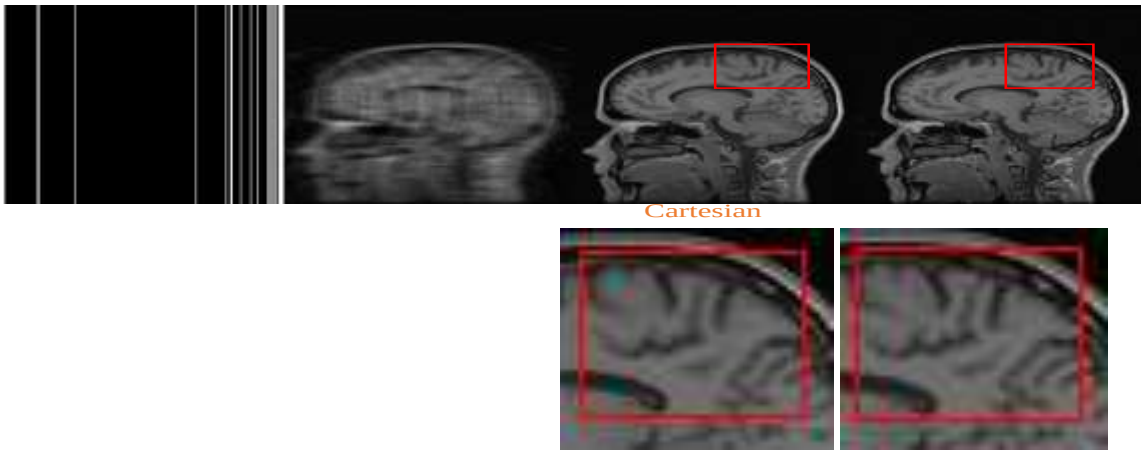
Zeng, Z., Xie, W., Zhang, Y., & Lu, Y. (2019). RIC-Unet: An Improved Neural Network Based on Unet for Nuclei Segmentation in Histology Images. *IEEE Access*, *7*, 21420–21428. https://doi.org/10.1109/ACCESS.2019.2896920

Zhang, H. M., & Dong, B. (2020). A Review on Deep Learning in Medical Image Reconstruction. *Journal of the Operations Research Society of China*, *8*(2), 311–340. https://doi.org/10.1007/S40305-019-00287-4/FIGURES/6

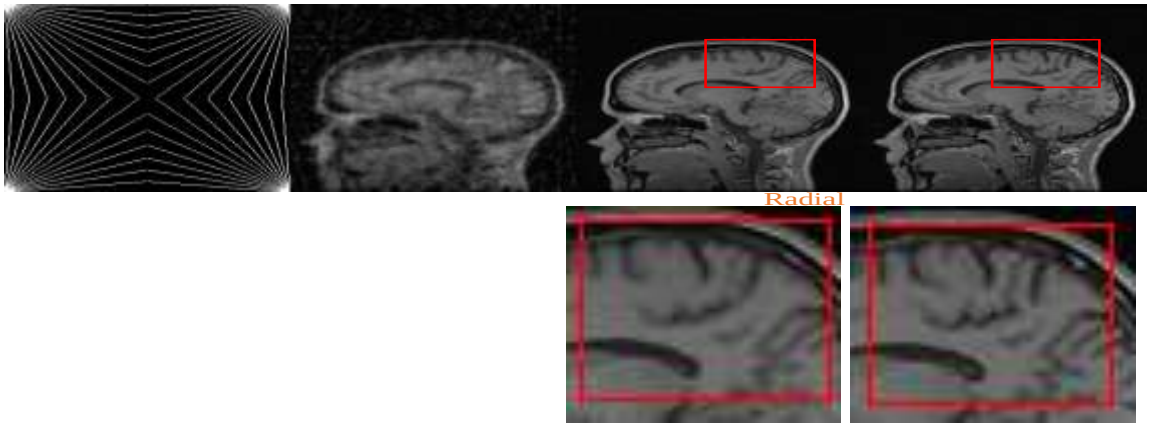
Zhang, Z., Liu, Q., & Wang, Y. (2018). Road Extraction by Deep Residual U-Net. *IEEE Geoscience and Remote Sensing Letters*, *15*(5), 749–753. https://doi.org/10.1109/LGRS.2018.2802944

Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. *Proceedings of the IEEE International Conference on Computer Vision*, *2017-Octob*, 2242–2251. https://doi.org/10.1109/ICCV.2017.244

Appendix A: Samples of Results by Zooming Images

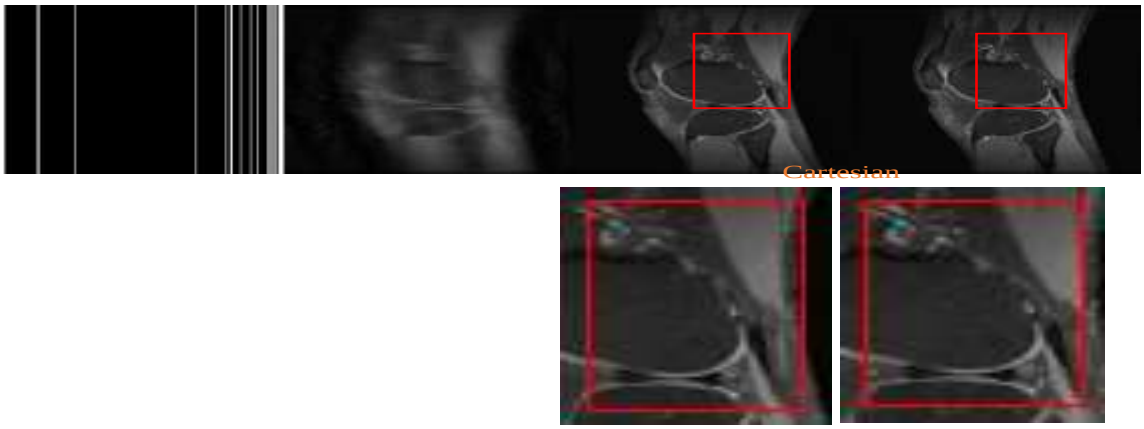


(a)

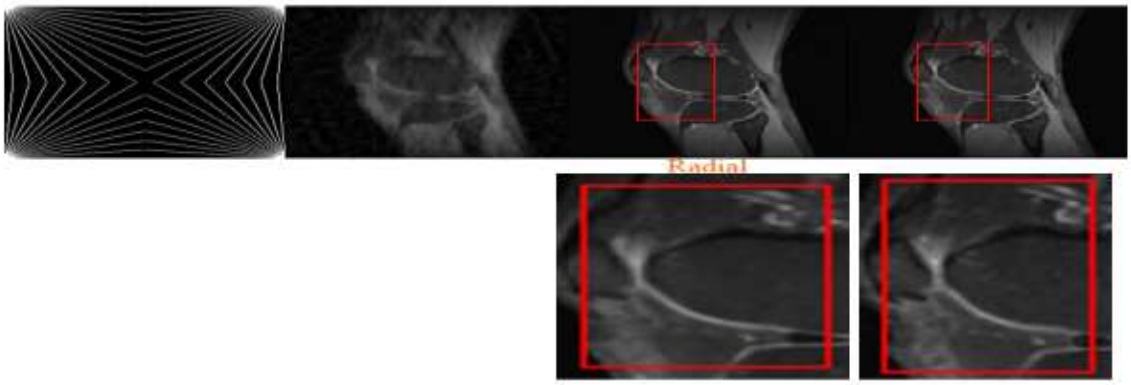


(b)

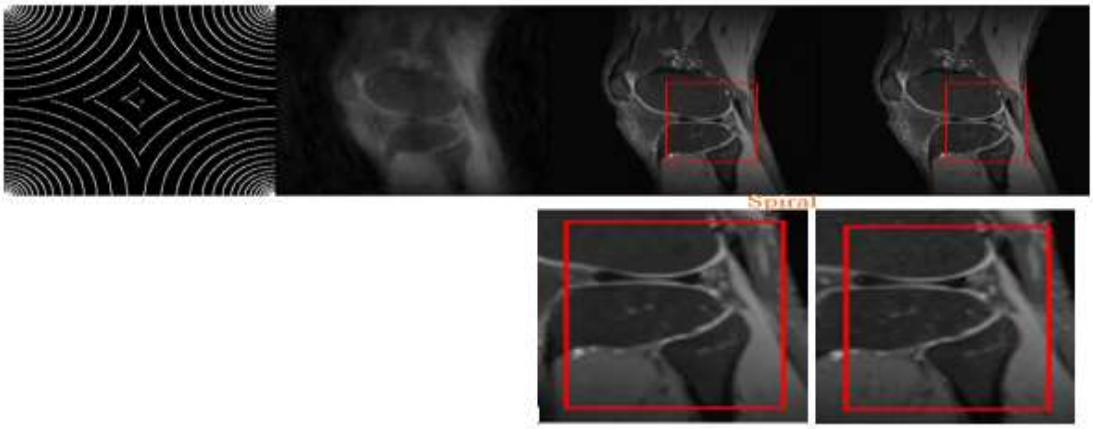
(c)



(d)



(e)



(f)

The above images (a), (b), and (c) were represents brain MRI reconstructed and ground truth with cartesian 12.5%, radial 10%, and spiral 10% respectively. (d), (e) and (f) also represent knee MRI reconstructed and ground truth with cartesian 12.5%, radial 10%, and spiral 10% respectively.

Appendix B: List of sample codes

Self-attention mechanism implementation:

```
class SelfAttention(nn.Module):
    def __init__(self, in_channels):
        super(SelfAttention, self).__init__()
        self.query_conv = nn.Conv2d(in_channels, in_channels//8,
kernel_size=1)
        self.key_conv = nn.Conv2d(in_channels, in_channels//8,
kernel_size=1)
        self.value_conv = nn.Conv2d(in_channels, in_channels,
kernel_size=1)
        self.gamma = nn.Parameter(torch.zeros(1))

    def forward(self, x):
        batch_size, C, H, W = x.size()
        proj_query = self.query_conv(x).view(batch_size, -1,
H*W).permute(0, 2, 1)
        proj_key = self.key_conv(x).view(batch_size, -1, H*W)
        energy = torch.bmm(proj_query, proj_key)
        attention = torch.softmax(energy, dim=-1)
        proj_value = self.value_conv(x).view(batch_size, -1, H*W)
        out = torch.bmm(proj_value, attention.permute(0, 2, 1))
        out = out.view(batch_size, C, H, W)
        out = self.gamma * out + x
        return out
```

Spatial attention mechanism implementation:

```
class SpatialAttention(nn.Module):
    def __init__(self, in_channels):
        super(SpatialAttention, self).__init__()
        self.conv = nn.Conv2d(in_channels=in_channels,
out_channels=1, kernel_size=1)
        self.sigmoid = nn.Sigmoid()

    def forward(self, x):
        avg = torch.mean(x, dim=1, keepdim=True)
        max = torch.max(x, dim=1, keepdim=True)[0]
        out = torch.cat([avg, max], dim=1)
        out = self.conv(out)
        out = self.sigmoid(out)
        return out
```

Channel-wise attention mechanism implementation:

```
class ChannelWiseAttention(nn.Module):
    def __init__(self, in_channels, reduction=16):
        super(ChannelWiseAttention, self).__init__()
        self.avg_pool = nn.AdaptiveAvgPool2d(1)
        self.max_pool = nn.AdaptiveMaxPool2d(1)
        self.fc = nn.Sequential(
            nn.Conv2d(in_channels, in_channels // reduction,
kernel_size=1, padding=0),
            nn.ReLU(inplace=True),
            nn.Conv2d(in_channels // reduction, in_channels,
kernel_size=1, padding=0),
            nn.Sigmoid()
        )

    def forward(self, x):
        avg_out = self.avg_pool(x)
        max_out = self.max_pool(x)
        out = avg_out + max_out
        out = self.fc(out)
        out = x * out
        return out
```

Residual U-Net implementation:

```
class Residual(nn.Module):
    def __init__(self, chan):
        super(Residual, self).__init__()
        self.block = nn.Sequential(
            nn.Conv2d(chan, chan, kernel_size=3, stride=1,
padding=1, bias=False),
            nn.BatchNorm2d(chan),
            nn.LeakyReLU(0.2),
            nn.Conv2d(chan, chan // 2, kernel_size=3, stride=1,
padding=1, bias=False),
            nn.BatchNorm2d(chan // 2),
            nn.LeakyReLU(0.2),
            nn.Conv2d(chan // 2, chan, kernel_size=3, stride=1,
padding=1, bias=False),  # nl=tf.identity
            nn.BatchNorm2d(chan),
            nn.LeakyReLU(0.2)
        )

    def forward(self, x):
        return x + self.block(x)


class Residual_enc(nn.Module):
    def __init__(self, in_chan, chan):
```

```
        super(Residual_enc, self).__init__()
        self.block = nn.Sequential(
            nn.Conv2d(in_chan, chan, kernel_size=3, stride=2,
padding=1, bias=False),
            nn.BatchNorm2d(chan),
            nn.LeakyReLU(0.2),
            Residual(chan),
            nn.Conv2d(chan, chan, kernel_size=3, stride=1,
padding=1, bias=False),
            nn.BatchNorm2d(chan),
            nn.LeakyReLU(0.2)
        )
        self.spa = SpatialAttention(chan)
        self.sa = SelfAttention(chan)

    def forward(self, x):
        return self.block(x)


class Residual_dec(nn.Module):
    def __init__(self, in_chan, chan):
        super(Residual_dec, self).__init__()
        self.block = nn.Sequential(
            nn.ConvTranspose2d(in_chan, chan, kernel_size=3,
stride=1, padding=1, bias=False),
            nn.BatchNorm2d(chan),
            nn.LeakyReLU(0.2),
            Residual(chan),
            nn.ConvTranspose2d(chan, chan, kernel_size=3,
stride=2, padding=1, output_padding=1, bias=False),
            nn.BatchNorm2d(chan),
            nn.LeakyReLU(0.2),
        )
        self.spa = SpatialAttention(chan)
        self.ca = ChannelWiseAttention(chan)

    def forward(self, x):
        return self.block(x)
```

Overall Generator Implementation:

```
class Generator(nn.Module):

    def __init__(self):
        super(Generator, self).__init__()

        self.e0 = Residual_enc(2, NB_FILTERS * 1)
        self.e1 = Residual_enc(NB_FILTERS * 1, NB_FILTERS * 2)
        self.e2 = Residual_enc(NB_FILTERS * 2, NB_FILTERS * 4)
        self.e3 = Residual_enc(NB_FILTERS * 4, NB_FILTERS * 8)
```

```
        self.d3 = Residual_dec(NB_FILTERS * 8, NB_FILTERS * 4)
        self.d2 = Residual_dec(NB_FILTERS * 4, NB_FILTERS * 2)
        self.d1 = Residual_dec(NB_FILTERS * 2, NB_FILTERS * 1)
        self.d0 = Residual_dec(NB_FILTERS * 1, NB_FILTERS * 1)

        self.dd = nn.Sequential(
            nn.Conv2d(NB_FILTERS, 2, kernel_size=3, stride=1,
padding=1),
            nn.Tanh()
        )

    def forward(self, x):
        x_e0 = self.e0(x)
        x_e1 = self.e1(x_e0)
        x_e2 = self.e2(x_e1)
        x_e3 = self.e3(x_e2)

        x_d3 = self.d3(x_e3)
        x_d2 = self.d2(x_d3 + x_e2)
        x_d1 = self.d1(x_d2 + x_e1)
        x_d0 = self.d0(x_d1 + x_e0)

        out = self.dd(x_d0)

        return out + x


class HARA_G(nn.Module):

    def __init__(self):
        super(HARA_G, self).__init__()
        self.g0 = Generator()
        self.g1 = Generator()
        self.dc = Dataconsistency()

        for m in self.modules():
            if isinstance(m, nn.Conv2d) or isinstance(m,
nn.ConvTranspose2d):
                nn.init.trunc_normal_(m.weight, mean=0, std=0.02,
a=-0.04, b=0.04)
                if hasattr(m, 'bias') and m.bias is not None:
                    nn.init.constant_(m.bias.data, 0.0)
            if isinstance(m, nn.BatchNorm2d):
                nn.init.uniform_(m.weight, a=-0.05, b=0.05)
                nn.init.constant_(m.bias, 0)

    def forward(self, x_un, k_un, mask):
        x_rec0 = self.g0(x_un)
```

```
        x_dc0 = self.dc(x_rec0, mask, k_un)
        x_rec1 = self.g1(x_dc0)
        x_dc1 = self.dc(x_rec1, mask, k_un)

        return x_rec0, x_dc0, x_rec1, x_dc1
```

Relative Average Discriminator implementation:

```
class RelativisticAvgDiscriminator(nn.Module):

    def __init__(self, nb_filters=64):
        super(RelativisticAvgDiscriminator, self).__init__()

        self.e0 = Residual_enc(2, nb_filters * 1)
        self.e1 = Residual_enc(nb_filters * 1, nb_filters * 2)
        self.e2 = Residual_enc(nb_filters * 2, nb_filters * 4)
        self.e3 = Residual_enc(nb_filters * 4, nb_filters * 8)

        self.ret = nn.Conv2d(nb_filters * 8, 1, kernel_size=4,
stride=1, padding=2)
        # initialize the weights and biases of the model
        for m in self.modules():
            if isinstance(m, nn.Conv2d) or isinstance(m,
nn.ConvTranspose2d):
                nn.init.trunc_normal_(m.weight, mean=0, std=0.02,
a=-0.04, b=0.04)
                if hasattr(m, 'bias') and m.bias is not None:
                    nn.init.constant_(m.bias.data, 0.0)
            if isinstance(m, nn.BatchNorm2d):
                nn.init.uniform_(m.weight, a=-0.05, b=0.05)
                nn.init.constant_(m.bias, 0)

    def forward(self, x):
        x_e0 = self.e0(x)
        x_e1 = self.e1(x_e0)
        x_e2 = self.e2(x_e1)
        x_e3 = self.e3(x_e2)
        ret = self.ret(x_e3)
        # apply average pooling to get the relative average
discriminator score
        avg_pool = nn.functional.avg_pool2d(ret,
ret.size()[2:]).view(ret.size()[0], -1)
        rel_avg_score = avg_pool / torch.norm(avg_pool, p=2,
dim=1, keepdim=True)

        return rel_avg_score
```