University of Padova

Department of Physics and Astronomy "Galileo Galilei"

Master Thesis in Physics Of Data

Synthetic CT Generation from MR Images : a U-Net Deep Learning Approach

Internal Supervisor
Prof.ssa Alessandra Bertoldo
University of Padova

External supervisors
Prof. Dr.-Ing. Maria Francesca Spadea
Karlsruhe Institute Of Technology, Germany

Prof. Paolo Zaffino
Magna Græcia University of Catanzaro, Italy

Master Candidate
Farshad Jafarpour

Academic Year

2023-2024

To my mother, ***Farideh*** ;
the Center of my world

Abstract

Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) work hand in hand in radiation therapy planning—CT delivers essential attenuation data for accurate dose calculations, while MRI excels at capturing detailed soft tissue contrasts, enhancing precision in treatment planning. However, repeated CT scans pose risks due to excess radiation exposure and higher costs. This thesis proposes a deep learning approach using a 2D U-Net model for generating synthetic CT (sCT) images from MRI data. While this approach shows promise in reducing dependence on CT, further advancements are essential before sCT can fully replace CT in treatment planning. The study focuses on neurodegenerative patients, whose imaging often suffers from artifacts due to involuntary movements. We develop a robust preprocessing pipeline using 3D Slicer, an open software in the medical imaging community, to achieve suitable rigid registration of CT and MRI images. The U-Net model was pre-trained on a high-quality dataset of Glioblastoma patients and then fine-tuned using transfer learning on the target dataset of neurodegenerative patients. We evaluate the model's performance using Mean Absolute Error (MAE) and Mean Error (ME), highlighting challenges posed by patient movement and metal artifacts. The results demonstrate the model's capability to generate accurate sCT images, though with varying performance depending on the quality of the input MRI. In the final analysis, this work underscores the potential of deep learning in reducing radiation exposure in radiotherapy but also highlights the need for further refinement in handling artifacts.

Contents

Listing of figures

Listing of tables

Listing of acronyms

AI	Artificial Intelligence
BN	Batch Normalization
CBCT	Cone Beam Computed Tomography
CNN	Convolutional Neural Network
CT	Computed Tomography
DL	Deep Learning
DSC	Dice Similarity Coefficient
MAE	Mean Absolute Error
ME	Mean Error
ML	Machine Learning
MRI	Magnetic Resonance Imaging
MRgRT	Magnetic Resonance Imaging Guided Radiotherapy
PET	Positron Emission Tomography
SGD	Stochastic Gradient Descent
sCT	Synthetic Computed Tomography
STD	Standard Deviation

1

Introduction

Deep learning is revolutionizing healthcare by enabling the analysis of complex, diverse medical data, like electronic health records, imaging, and genomics. Unlike traditional methods requiring manual feature extraction, deep learning models learn directly from raw data, offering better predictive accuracy. Despite its capacity for precision medicine, challenges persist due to the heterogeneous and noisy nature of biomedical data, as well as the requirement for interpretable models in clinical practice. While deep learning holds great promise for precision medicine, further advancements are needed for widespread clinical integration[3]. The role of deep learning in Image synthesis research is quite thrilling, involving the conversion of one imaging modality into another. The advantages of synthesis encompass the prevention of potential artifacts, lowering patient expenses and discomfort, and eliminating exposure to radiation[4]. MRI-based sCT is the most extensively researched and influential application of synthesis models in radiation therapy, offering the potential to replace traditional CT in treatment planning due to its superior soft tissue contrast and the ability to avoid the radiation dose associated with CT. Although MRI is excellent for imaging soft tissue, it does not provide the attenuation data required for precise dose calculations generally obtained from CT scans. Therefore opting for MR-based sCT can avoid this obstacle. Despite its advantages, MRI-based sCT faces challenges, particularly in accurately reconstructing bone and air, which have low proton density and weak signals, leading to potential errors[5].

1.1 GOALS

Deep learning models are highly sensitive to the quality and quantity of the data they process [6]. In previous research, Spadea et al.[2] utilized synthetic CT generation for glioblastoma patients.Building on this, we used the same dataset as a baseline to train our model, saving the model weights to apply them to a target dataset of neurodegenerative patients. This new dataset was of similar size but had lower image quality due to involuntary movements common in these patients.The main objective of this thesis is to evaluate the performance of a machine learning model for synthetic CT generation when trained on a high-quality baseline dataset versus datasets affected by artifacts.

1.2 THESIS STRUCTURE

The thesis is organized into several key sections following the goals of the study. The **Background** chapter provides a comprehensive overview of the theoretical context and literature relevant to medical image translation. The **Method** section details the data preprocessing techniques, model architecture, and training processes employed in the research. The **Results** chapter presents the outcomes of the experiments, including error metrics and visualizations. The thesis concludes with a **Conclusion** that discusses the implications of the findings and potential future work.

2

Background

2.1 PREVIOUS WORKS

Several studies have explored the use of deep learning (DL) models, such as convolutional neural networks (CNNs) and generative adversarial networks (GANs), for synthetic image generation, particularly in the context of synthetic computed tomography (sCT) from magnetic resonance imaging (MRI). Early applications of CNNs involved convolutional encoder-decoder (CED) architectures and their variants such as U-Net [7], ResNet [8], SE-ResNet [9], and DenseNet [10]. These models have demonstrated effectiveness in various medical image translation tasks by leveraging their feature extraction capabilities, especially the encoder-decoder structure of U-Net, which excels in preserving spatial information through its skip connections [7].

Generative adversarial networks (GANs), first introduced by Goodfellow et al. [11], further enhanced the performance of image generation models. The GAN framework employs two networks: a generator that creates synthetic images and a discriminator that evaluates their realism. The U-Net architecture is often used as the generator in these GAN models, as demonstrated in studies by Emami et al. [12], who used a U-Net-based GAN to generate sCT from MRI. Another popular approach involves the use of conditional GANs (cGANs), where both generator and discriminator are conditioned on input images, resulting in more accurate sCT predictions [12, 13, 14].

One notable application of GANs in the field is the use of cycle-GANs, which are designed to perform image-to-image translation tasks without the need for paired training data. Studies by Gong et al. [15] and Xue et al. [16] explored cycle-GANs to translate between MRI and sCT modalities. These studies showed that cycle-GANs could generate accurate sCT images with comparable performance to GANs trained on paired datasets, despite the challenges of unregistered data.

Recent advancements have also focused on the use of advanced loss functions, such as adversarial loss and structural similarity index (SSIM) loss, to improve the visual quality of synthetic images [14]. These functions guide the model to produce images that are not only visually similar to ground truth but also maintain fine structural details critical for clinical applications.

In summary, the integration of CNNs and GANs has significantly advanced the field of synthetic CT generation from MRI. U-Net, in particular, has emerged as a preferred architecture for many models due to its ability to efficiently extract and reconstruct image features. The development of cGANs and cycle-GANs has further improved model accuracy, even when trained on unpaired datasets. These contributions have laid a solid foundation for future research in synthetic imaging and its applications in radiation therapy and diagnostic radiology.

2.2 Medical Image translation Applications

2.2.1 Radiation Therapy planning

Radiation therapy is a primary treatment for cancer, used in more than half of all cancer cases, with treatment planning being a crucial and resource-intensive part of the process. Over the past few decades, advancements in operations research and computer science have made treatment planning increasingly sophisticated, allowing for the creation of complex radiation therapy plans that aim to minimize damage to healthy tissue while maximizing tumor control[17]. Treatment planning in radiotherapy commonly involves using MR and CT imaging for many patients. CT images are used to calculate the dose and position the patient before treatment, providing electron density maps and reference images. MR images provide excellent tissue contrast for diagnosing tumors and organs at risk. Through image registration, MR contours are propagated to CT images for treatment planning. However, combining both modalities leads to systematic image fusion errors, adding to the time and expense costs for the patient. Additionally, the use of CT may expose patients to non-negligible doses of ionizing radiation[18]. MR-guided radiotherapy (MRgRT) utilizes MRI for its superb soft tissue contrast, enabling precise tumor targeting. However, MRI lacks the necessary tissue density information for accurate dose calculations, which is typically provided by CT scans. Synthetic CT (sCT) steps in to address this gap by generating the missing attenuation information from MRI data. This enables accurate dose planning by combining the benefits of MRI's superior imaging with CT's essential dose calculation capabilities, ultimately enhancing the precision and effectiveness of the treatment plan while minimizing the need for additional CT scans and associated radiation exposure[19].

2.2.2 PET Attenuation Correction

In PET imaging, attenuation correction is vital for ensuring accurate quantification of radiotracer distribution. As the PET signals pass through the body, tissues can absorb or scatter photons, leading to inaccuracies in the detected signal. To correct for this, attenuation maps, typically derived from CT images, are used to adjust the PET data, ensuring that the final images accurately represent the true tracer concentration. This process is critical for both diagnostic accuracy and subsequent treatment planning[20]. Traditionally, attenuation maps are derived from CT images due to their ability to provide the necessary density information. However, in PET/MRI systems, where CT data isn't available, synthetic CT (sCT) images generated from MRI data can be used to create these attenuation maps. sCT provides the required attenuation information, allowing for accurate PET attenuation correction in MRI-based workflows, ensuring the high-quality imaging needed for effective patient care[21].

2.3 Deep Learning in Medical Image translation

2.3.1 Machine Learning and Neural Networks

Before discussing deep learning, it is crucial to establish a foundational understanding by distinguishing key concepts in the field of artificial intelligence (AI).

AI, machine learning (ML), and deep learning (DL) are interrelated, and their hierarchical relationship is often depicted in a Venn diagram 2.1, illustrating how these fields overlap and build upon one another.

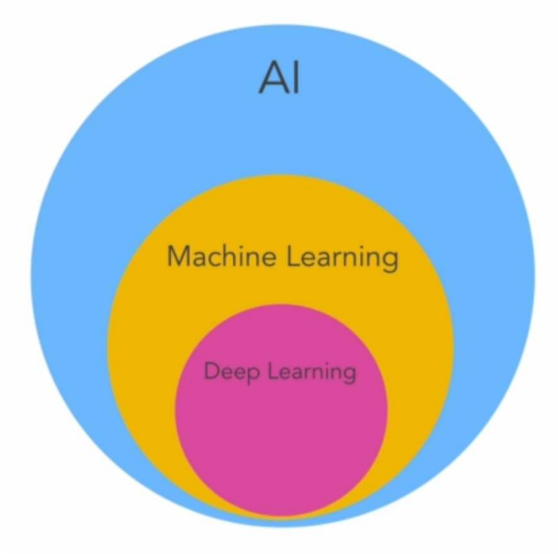


Figure 2.1: The hierarchical relationship between AI, ML and DL.

Artificial Intelligence (AI) refers to the creation of systems capable of tasks that typically require human intelligence, such as decision-making and language comprehension.The ”holy grail” of AI in this century is the pursuit of Artificial General Intelligence **(AGI)**, which would exhibit human-like cognitive abilities across diverse tasks. A subset of AI, **Machine Learning (ML)**, involves algorithms that learn from data, enabling computers to make predictions or decisions without explicit programming.

Deep Learning (DL), a more specific subset of ML, employs deep neural networks to automatically identify intricate patterns in extensive datasets, performing well in tasks such as image and speech recognition.Deep Learning (DL), a more specific subset of ML, employs deep neural networks to automatically identify intricate patterns in extensive datasets, performing well in tasks such as image and speech recognition[22].

The evolution of deep learning is rooted in concepts inspired by human cognition and biological neural networks. Initially, artificial neural networks (ANNs) were designed to mimic how neurons work in the brain, with the simplest model being the perceptron—a single-layer network capable of handling linear functions through activation functions like ReLU and sigmoid[23].A single perceptron calculates an output y using the formula:

$$y = \phi\left(\sum_{i=1}^{n} w_i x_i + b\right)$$

where x_i represents the input features, w_i the corresponding weights, b the bias term, and ϕ the activation function.

Although single-layer perceptrons 2.2 were capable of approximating simple functions, the incorporation of multiple layers in multi-layer perceptrons (MLPs) greatly broadened their capacity to depict intricate patterns.

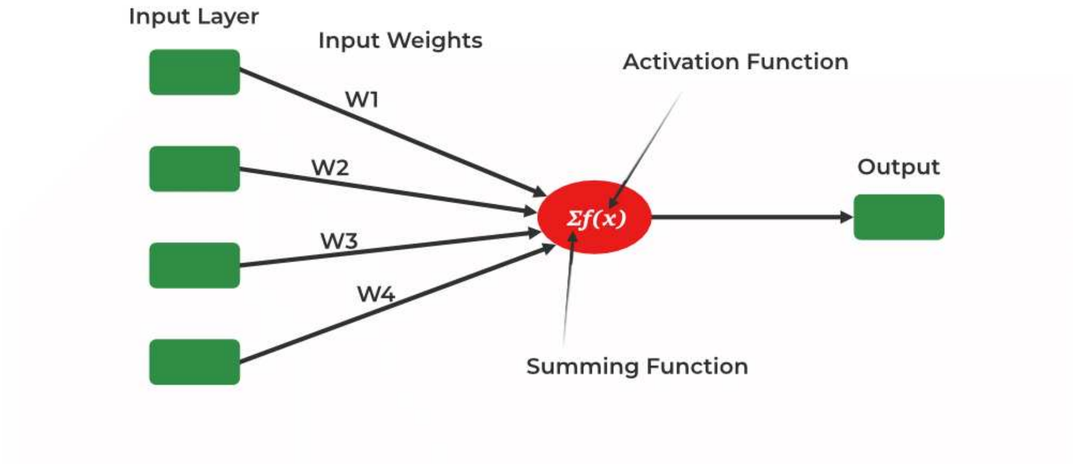


Figure 2.2: A single layer perceptron model

This development gave rise to deep neural networks (DNNs) equipped with numerous hidden layers, signifying the transition to deep learning[24]. Neural Networks 2.3 consist of different layers of neurons organized as follows:

- **Input Layer**: This layer is made up of neurons that take in input data from external sources.
- **Hidden Layers**: These layers contain neurons that process the input data by transforming it through weighted connections and activation functions. They are not directly exposed to the external environment.
- **Output Layer**: Neurons in this layer provide the final output of the network, which is then delivered back to the environment.

Neurons in a neural network are connected to neurons in adjacent layers, either fully or partially, depending on the architecture of the network and the specific task it is meant to accomplish[25].

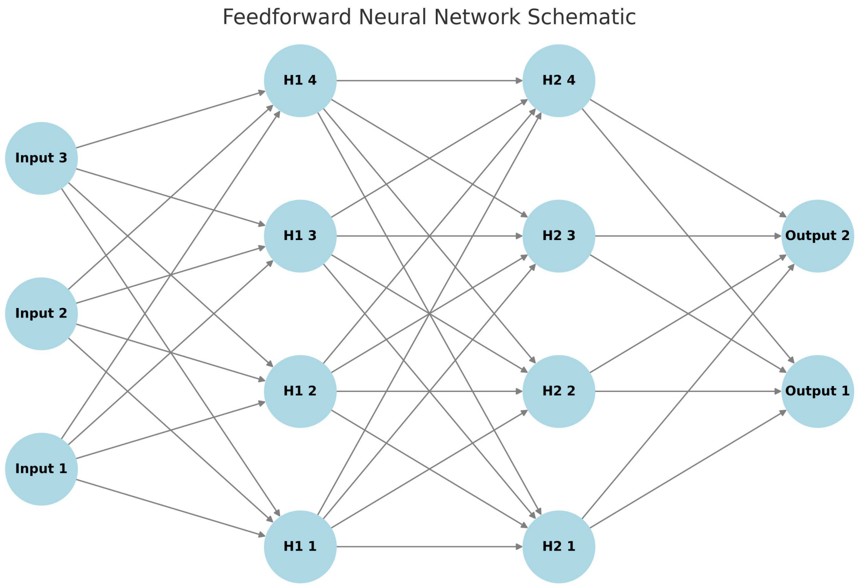


Figure 2.3: Example of how a Neural Network works.

Learning in a neural network involves two main phases:

1. **Forward Propagation**: Input data is fed into the input layer and moves through the hidden layers, finally reaching the output layer where a prediction is made.
2. **Backpropagation**: If there's a discrepancy between the predicted output and the actual target, an error is calculated and sent back through the network. The weights are then adjusted to reduce this error in future predictions.

Machine learning techniques are mainly categorized into four types: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning[26].

- **Supervised Learning:** This technique involves learning a function that maps inputs to outputs based on labeled training data. It is a goal-oriented approach where the model learns from input-output pairs to make predictions. Common tasks include classification (e.g., text sentiment analysis) and regression, where the objective is to separate data or fit it to a specific model.
- **Unsupervised Learning:** In this approach, the model analyzes unlabeled data without human intervention, aiming to uncover hidden patterns, structures, and relationships within the dataset. Unsupervised learning is often used for tasks such as clustering, feature learning, dimensionality reduction, anomaly detection, and exploratory analysis.

- **Semi-Supervised Learning:** It is a blend of supervised and unsupervised learning, working with both labeled and unlabeled data. It is especially useful when labeled data is scarce and expensive to obtain, while large amounts of unlabeled data are available. Application areas include machine translation, fraud detection, and text classification.

- **Reinforcement Learning:** Reinforcement learning involves agents learning optimal actions within an environment based on a reward-punishment system. The goal is to maximize rewards or minimize risks through continuous interaction with the environment. This method is effective for complex tasks like robotics, autonomous driving, and optimizing logistics systems but is less suited for straightforward problems.

2.3.2 CONVOLUTIONAL NEURAL NETWORKS

A Convolutional Neural Network **(CNN)** is a deep learning model inspired by the visual cortex of animals, designed to process grid-like data, such as images[27]. It automatically learns spatial hierarchies of features through multiple layers.

As illustrated in Figure 2.4, CNNs begin with feature extraction using **convolution layers** where filters (kernels) slide over the input image. Stride and padding are crucial here; stride controls the movement of the filters, while padding ensures complete coverage of the input matrix by adding zeros as needed. These convolution layers detect various features, which are then passed through **pooling layers** (also known as downsampling). Pooling layers reduce the dimensionality of the feature maps by applying a filter that aggregates the values within its receptive field. Unlike convolutional layers, pooling layers do not have weights; instead, they use functions like max pooling, which selects the maximum value, or average pooling, which calculates the average value within the receptive field[28]. Although pooling results in some information loss, it significantly reduces the complexity of the network, improves computational efficiency, and helps prevent overfitting.

The output from these layers is passed through **classification** stages involving fully connected layers. These layers combine and refine the extracted features, leading to a final prediction or classification of the input. Throughout this process, the CNN becomes increasingly adept at recognizing complex patterns within the data[29]. Training of these networks involves optimizing parameters, such as the weights of the kernels, using techniques like backpropagation, ensuring the CNN's accuracy in tasks like image recognition.

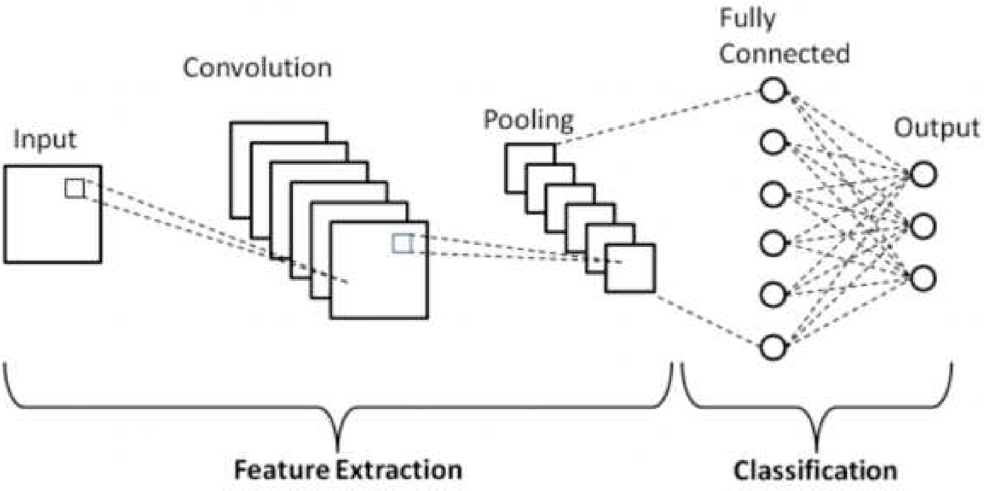


Figure 2.4: Schematic diagram of a basic CNN architecture

Dropout is a regularization technique used in deep neural networks to prevent overfitting. By randomly dropping units and their connections during training, Dropout reduces the network's tendency to over-rely on specific features. This forces the network to learn more robust, generalized patterns, leading to improved performance on various tasks[1].

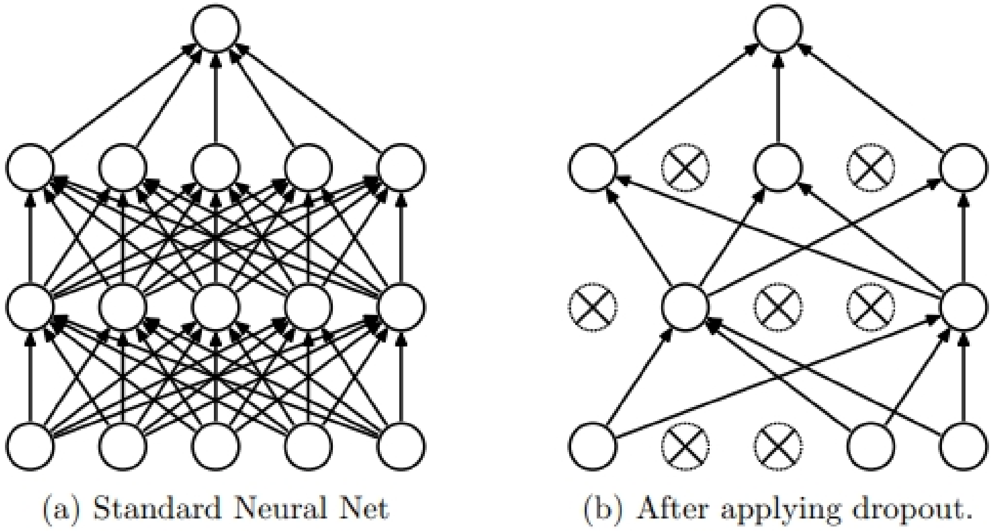

(a) Standard Neural Net
(b) After applying dropout.

Figure 2.5: Drop out Neural Network Model from [1]

Batch Normalization (BN) is a key technique in Convolutional Neural Networks (CNNs) that addresses internal covariate shift, where layer input distributions change during training. BN normalizes activations within each mini-batch, leading to more stable and efficient training. This allows for higher learning rates, faster convergence. By stabilizing the learning process, BN not only speeds up training but also enhances model performance[30].

2.3.3 TRAINING

Imagine you're climbing down a mountain at night with only a dim flashlight. To find your way down, you look around and choose the direction where the ground slopes the most steeply downward, stepping in that direction repeatedly. If you get stuck in a small dip, you follow your momentum to get out. This gradual, iterative approach will eventually get you to the base camp.

This scenario is similar to training a neural network using a technique called **gradient descent**. In neural networks, training involves finding the best set of weights to maximize accuracy. Gradient descent helps achieve this by iteratively adjusting the weights in the direction that most improves performance, much like your strategy of stepping downhill[31].

Once the dataloader is ready , the model can be trained using the training set, a **loss function** and an optimization algorithm . The loss function quantifies the difference between the predicted output and the actual target. During training, the network adjusts its weights to minimize this loss. Common loss functions include mean squared error **(MAE)** for regression tasks and **cross-entropy** loss for classification problems[25].

Optimization algorithms like **(SGD)** are techniques used to minimize a loss function by adjusting the model's parameters. They iteratively update the model's weights based on the gradient of the loss function to improve the model's performance.

Regarding the **hyperparameters** we are dealing with some elements in the training phase such as :

- **Learning Rate:** The learning rate controls how much the model's weights are adjusted with respect to the gradient during each iteration. If the learning rate is too high, the model may converge too quickly to a suboptimal solution. If it's too low, the training may be slow or get stuck in local minima. Optimal learning rate selection is crucial for effective training[25].

- **Epochs:** An epoch refers to one complete pass through the entire training dataset. During training, the model's weights are updated after each batch (or mini-batch), and the process is repeated for a specified number of epochs to optimize the network. Increasing the number of epochs allows the model to learn better, but it may lead to overfitting if set too high[32].

- **Lambda (Regularization Parameter): Lambda (λ)** is a regularization parameter used to control overfitting by penalizing large weights. Common regularization techniques include L2 regularization (Ridge) and L1 regularization (Lasso), where λ determines the strength of the penalty. A higher λ shrinks the weights more aggressively, which can prevent overfitting but might lead to underfitting[33].
- **Batch Size:** Batch size is the number of samples processed before the model's internal parameters are updated. Smaller batch sizes offer more accurate gradient estimates and introduce regularization through noise, while larger batch sizes speed up training by processing more data in parallel. The trade-off lies in balancing computation time and model performance[34].

3

Method

3.1 THE DATASET

3.1.1 THE BASELINE DATASET

The baseline dataset used in this study consisted of 15 pairs of T1-weighted MRI (T1MR) and CT image volumes, previously utilized in other research studies[2]. These images were acquired with patients in the supine position, all of whom were treated with radiation therapy following the surgical resection of glioblastoma. Ethical considerations were adhered to, with all patients providing written informed consent, and the study received approval from the local institutional review board.

MRI Data Acquisition: The MRI scans were performed using a 3 Tesla MAGNETOM Trio scanner (Siemens Healthcare GmbH, Erlangen, Germany). The scans had a voxel size of 1 × 1 × 1 mm^3, with a reconstruction matrix of 256 × 256 × 176. The scanned region covered the entire head down to the C3 vertebra, except in 5 patients, where the field of view (FOV) ended at the base of the skull. The MRI sequence parameters included a 3-dimensional acquisition with a gradient recalled inversion recovery sequence, a flip angle of 7°, echo time (TE) of 1.64 ms, repetition time (TR) of 2530 ms, inversion time (TI) of 1200 ms, pixel bandwidth of 650 Hz, number of frequency encoding steps (Nf) of 256, and a readout bandwidth of 166.4 kHz. The MRI scans were acquired using an in-house developed transmission and receiver coil (8-channel) designed for a simultaneous brain PET/MR scanner called brainPET.

To ensure patient comfort during the MRI, particularly given that each session lasted over an hour, a facial mask was not used to restrict head movement. This decision was also motivated by MRI safety concerns, as the radiofrequency currents generated by the MRI scanner could potentially cause overheating of the facial mask. Instead, patient head position was stabilized within the magnetic resonance coil using comfortable pads.

CT Data Acquisition: CT scans were performed using a LightSpeed QX/i scanner (GE Healthcare), with voxel sizes ranging from 0.49 × 0.49 × 2.5 mm^3 to 0.67 × 0.67 × 2.5 mm^3. During the CT scans, a facial mask was used to immobilize the patients' head positions to ensure consistency for treatment purposes. These CT images were subsequently used to train the network and served as ground truth (GT) for testing purposes. The interval between the T1MR and CT acquisitions ranged from 11 to 20 days.

Preprocessing: The preprocessing of the dataset involved masking out the patient head using Plastimatch software, setting the background image intensities to 0 for T1MR data and -1000 for CT data. A rigid transformation was computed to register the CT onto the T1MR, with Mattes Mutual Information as the metric to optimize the registration process. The final image resolution used for training was consistent with that of the MRI scans. Additionally, the T1MR images were tissue normalized using FreeSurfer, and the full range of intensities was preserved during the normalization process.

This baseline dataset, characterized by high-quality MRI and CT images, provided a robust foundation for training the U-Net model. The careful acquisition and preprocessing of this data were crucial in ensuring the model's ability to generate accurate synthetic CT images, which was later fine-tuned using the neurodegenerative patient dataset.

3.1.2 The Target Dataset

The target dataset used in this study consisted of MRI and CT image volumes acquired from a homogeneous group of 22 patients, comprising 13 females and 9 males. Unlike the baseline dataset, this cohort included patients with neurodegenerative conditions, which introduced specific challenges such as motion artifacts and metal-induced artifacts. These challenges significantly impacted the quality of the images and the performance of the synthetic CT (sCT) generation model.

MRI Data Acquisition: The MRI scans were obtained using a SIEMENS Biograph mMR scanner, which is a hybrid PET/MRI system designed for simultaneous PET/MR imaging. The MRI sequence used was Magnetization Prepared Rapid Gradient Echo (MP-RAGE), a

T1-weighted sequence known for its high contrast resolution, particularly useful in neuroimaging. The voxel size of the MRI scans was 1 × 1 × 1 mm^3 with an acquisition matrix of 256 × 248, ensuring detailed volumetric resolution. The scanning parameters included a slice thickness of 1 mm, a repetition time (TR) of 2300 ms, an echo time (TE) of 2.34 ms, and an inversion time (TI) of 900 ms. The flip angle was set at 8°, and the magnetic field strength was 3 Tesla. The MRI scans were acquired using a body coil for signal transmission, and the pixel bandwidth was 190 Hz. The percentage of phase field of view (FOV) was set at 96.875 percent, with 247 phase encoding steps.

CT Data Acquisition: The CT images were acquired using either a GE Discovery ST or GE Lightspeed scanner. Both scanners are commonly used in clinical settings for producing high-resolution images suitable for treatment planning. The challenges associated with neurodegenerative conditions—such as involuntary patient movements during scanning—necessitated extensive preprocessing to mitigate motion and metal-induced artifacts. The scans were initially reconstructed with a slice thickness of 1.25 mm and included detailed structural information critical for the sCT generation process. Despite preprocessing efforts, a few cases had to be excluded from training due to severe artifacts that could not be corrected.

This dataset, while more challenging than the baseline, was essential for fine-tuning the model to perform in less ideal, real-world conditions. The diversity in patient characteristics and imaging conditions made it a valuable test for assessing the model's robustness and generalizability.

3.2 PREPROCESSING FOR BOTH DATASETS

3.2.1 Z-SCORE NORMALIZATION

Given the differences in MRI acquisition protocols and the hardware used between the baseline and target datasets, it was crucial to address the variability introduced by the different scanners. Specifically, the baseline dataset was acquired using a 3 Tesla MAGNETOM Trio scanner, while the target dataset was obtained using a SIEMENS Biograph mMR, a hybrid PET/MRI system. These systems differ not only in their primary functions (e.g., the PET component of the Biograph mMR) but also in factors like voxel size, acquisition matrix, bandwidth, and scanning parameters. To account for these discrepancies and ensure consistency, we applied Z-score normalization using ***MONAI's NormalizeIntensity*** function. This standardization was necessary to align the intensity distributions across both scanners, facilitating reliable comparison and analysis of the MRI data across the two datasets.

Z-score Normalization:

Z-score normalization adjusts the intensity values of an image relative to the mean and standard deviation:

$$\text{normalized\_value} = \frac{\text{original\_value} - \text{mean\_value}}{\text{std\_value}}$$

This method preserves the tissue alignment in the images. While it works well when data comes from the same source, variations such as different vendors or scanner field strengths can affect voxel intensity. These variations can lead to poor model generalization across datasets acquired at different centers, making normalization essential for consistency[35].

3.3 Preprocessing for the target dataset

Data preprocessing of medical images in our target dataset, primarily utilizing ”3D Slicer”, an open-source software for medical imaging analysis, culminating in the registration of CT images to MR images. A significant challenge encountered during preprocessing was the presence of head immobilizers in the CT images, closely adjacent to the patient's head. This proximity made direct masking of the CT images to eliminate the head immobilizer impractical, complicating the registration process and making it time-consuming to achieve proper alignment for effective comparison and analysis.

The preprocessing workflow for CT images included these steps: foreground masking, scalar masking, cropping, and rigid registration. On the other hand, for MR preprocessing instead, we did bias field correction following foreground masking. Finally, by aligning the foreground MR mask with the registered CT scans we get the registered CT on MR, serving as the ground truth in our training phase. This comprehensive approach ensures the accuracy and reliability of the data used for further analysis and model training.

3.3.1 Bias Field Correction

In medical imaging, images should ideally exhibit a piecewise constant nature, where the intensity of the same tissue type remains consistent throughout an image, regardless of its location. This uniformity is crucial for automated analysis methods like segmentation, classification, and registration [36]. However, in practice, images often suffer from biased field artifacts due to

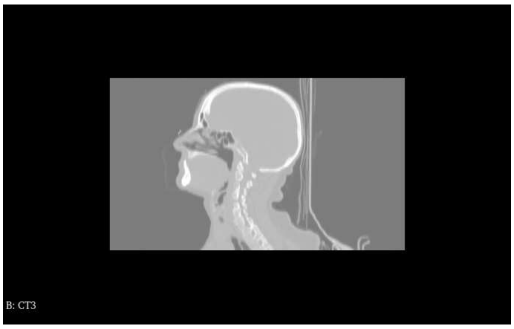


Figure 3.1: A slice of CT image from sagittal view

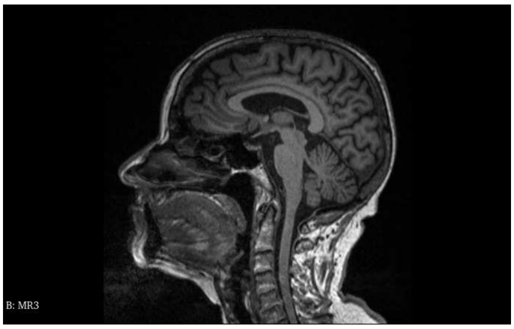


Figure 3.2: A slice of MR image from sagittal view

imperfections in the image acquisition. This artifact, manifesting as smooth intensity variations across the image, disrupts the tissue intensity uniformity, complicating automated analyses. For MRI Bias Field Correction, we utilized the N4ITK tool in the Filtering section of our process. We select our biased MR image as the input in the IO Section while maintaining the default settings for both the N4 parameter and advanced parameters. By clicking on the 'Apply' icon, we obtain the Bias Field Corrected MR image, which is then accessible in the 'SAVE' section, streamlining the correction process and enhancing the quality of the MR images for further analysis.

3.3.2 Foreground Mask

Foreground masking is vital for separating the skull from surrounding structures (the head immobilizer in CT cases). Therefore, we use the specialized segmentation extension in 3D Slicer. For the next step, in the IO section, we set the Output mask as 'CreateNewLabelMapVolume': as this facilitates subsequent analysis where the label map is essential for mask application. The 'Otsu Correction Factor' is responsible for the separation degree we need from surroundings so experimental adjustment of this factor in the Configuration Parameters is critical since it differs from one case to another, with a delicate balance required; a slight adjustment can significantly alter the defined volume of interest. For CT images, an average "0.1" correction factor was employed. Despite this, a minimal portion of the head immobilizers remained, necessitating further processing steps.

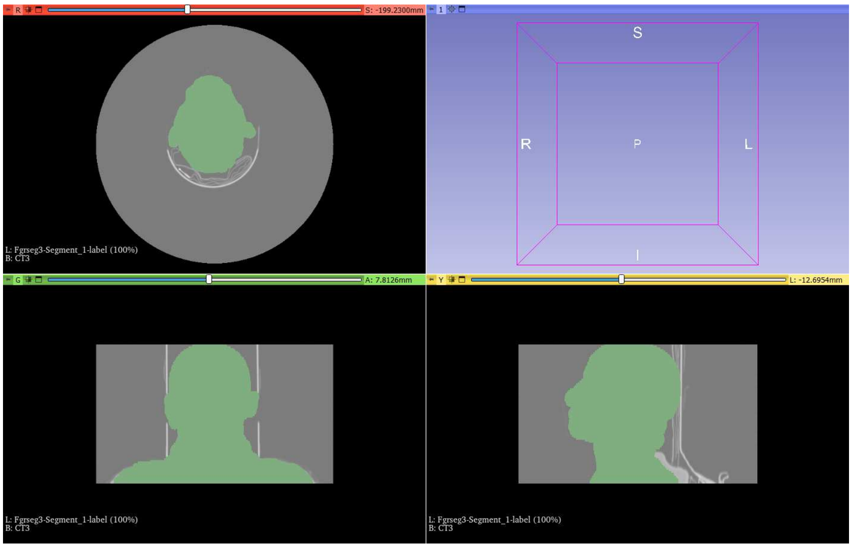


Figure 3.3: Foreground Label Maps

3.3.3 SCALAR MASK VOLUME

As we can see in Figures (3.1 and 3.2) both MR and CT scans have different background colors. Consequently, to harmonize CT images with MR images before registration, we use the Scalar Mask Volume extension in 3D Slicerto to modify the CT background. By applying this extension with the foreground mask obtained from the previous step, the CT image background becomes consistently dark, similar to MR images. This step is crucial for ensuring that the backgrounds of both modalities match, facilitating more accurate registration between them.

3.3.4 CROP VOLUME

As it is clear in figure 3.1, our CT images besides the skull contain part of the back of the neck that we don't have in MR cases. Thus, we can use the Crop Volume extension in 3D Slicer to adjust the field of view in CT images, ensuring they match the scale of MR images. This adjustment is essential for aligning the regions of interest across both modalities, facilitating accurate registration. By cropping the CT images, we achieve a zoomed-in effect that aligns with the dimensions and detail level of the corresponding MR images.

3.3.5 RIGID REGISTRATION

This step aligns MRI and CT data sets with high accuracy, ensuring anatomical congruence between modalities, a prerequisite for the creation of synthetic CT images that faithfully replicate the anatomical structures. Moreover, such precise alignment enables the utilization of evaluation metrics, notably the DICE coefficient and the mean absolute Hounsfield Unit (HU) intensity difference, to rigorously assess the synthetic image generation's anatomical and intensity fidelity, thereby affirming the method's reliability and effectiveness[37]. In the registration process using 3D Slicer's General Registration (BRAINS), MR images are set as the Fixed Image Volume and CT images as the Moving Image Volume. The 'use center of Head align' option is selected for Transform initialization, aiming for initial alignment based on the head's center. The registration phase is configured for 6 degrees of freedom (DOF) under Rigid settings, with other parameters left at default. Some artifacts from the head holder are still visible, necessitating further masking using MR images to refine the outcome.

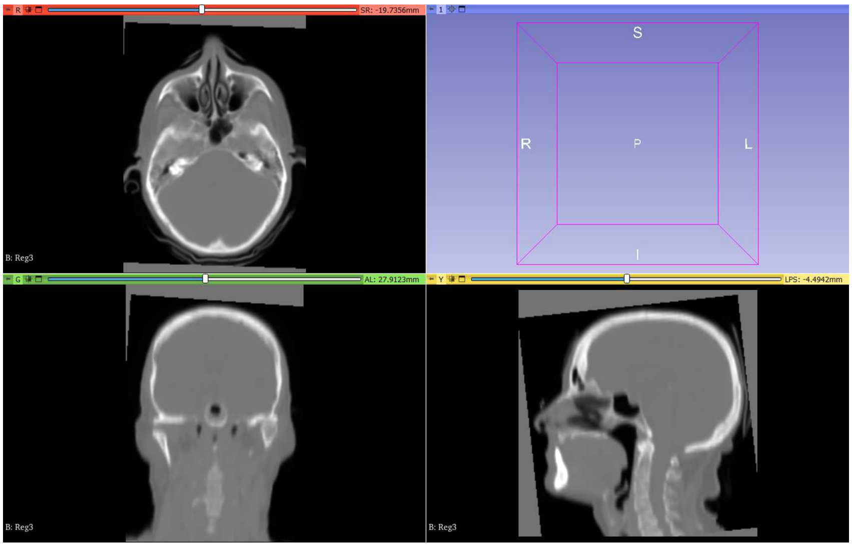


Figure 3.4: Registered Image, the artifact from head holder is still visible

3.3.6 MANUAL REGISTRATION

In the concluding phase of preprocessing, the foreground masking process, previously applied to MR images, is replicated on registered CT images to ensure accurate overlay. This critical step involves using 3D Slicer's Transforms extension, where the MR-derived foreground mask

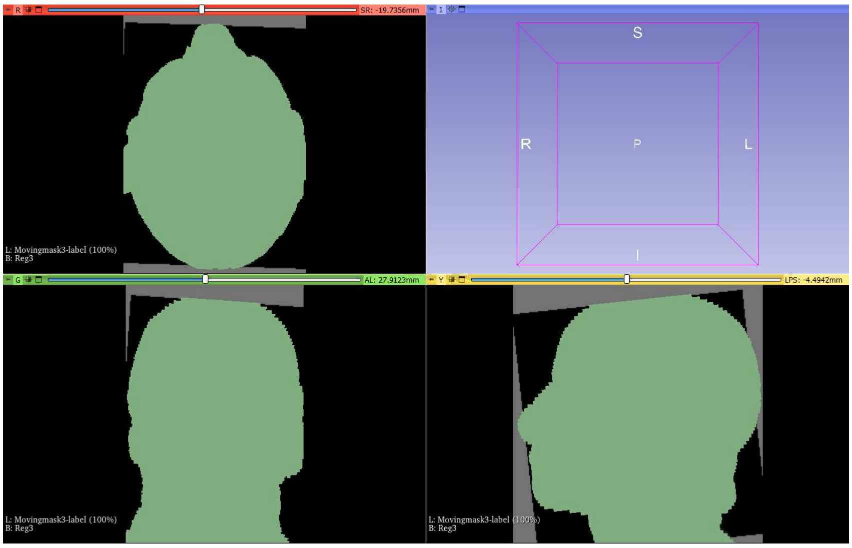


Figure 3.5: Applying MR mask on registered image

is selected for application. Through careful manipulation of translation and rotation controls, the mask is meticulously aligned with the registered CT image. The precise alignment is important for the success of subsequent analysis, ensuring that the transformation applied results in a finely adjusted, accurately overlaid image conducive to high-quality data interpretation for training phase.

3.3.7 WORKFLOW

For a better encapsulation , we have the whole preprocessing workflow in figure 3.6. It shows all steps one by one regarding relationships between each step in CT and MR.

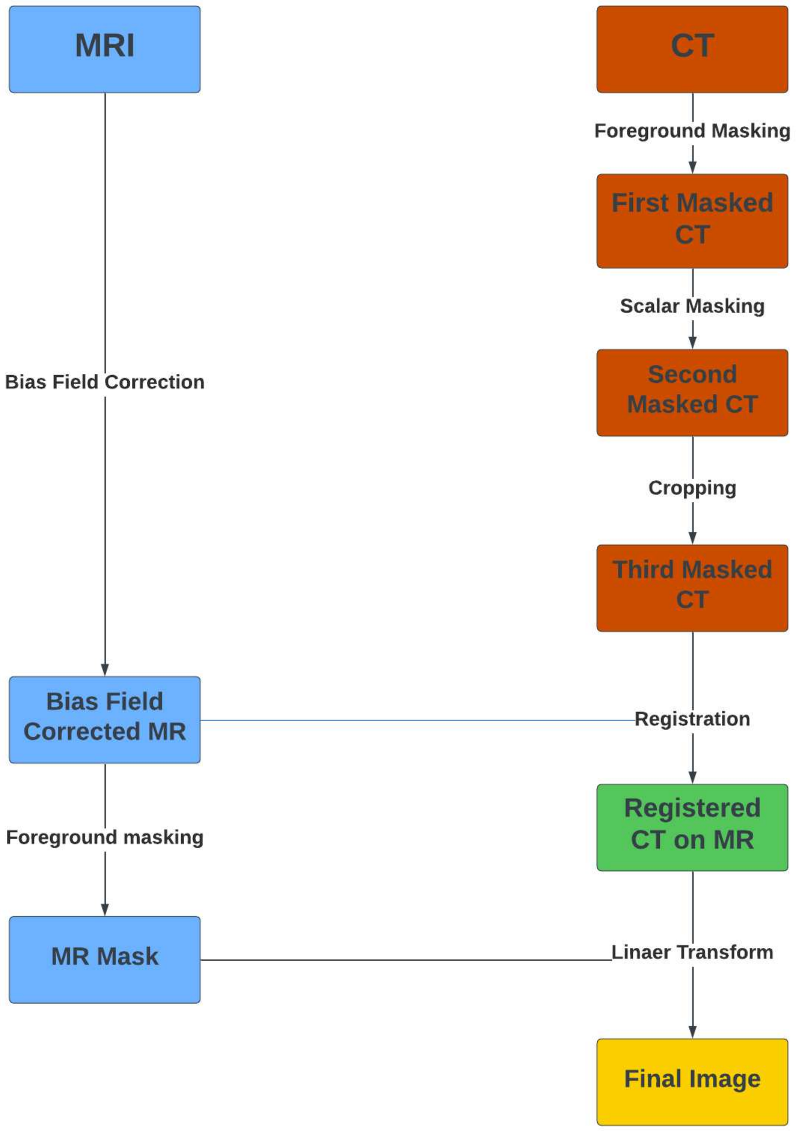


Figure 3.6: Preprocessing workflow

3.4 DATA AUGMENTATION

3.4.1 WHY DO WE NEED DATA AUGMENTATION?

The restricted access to training data presents a significant obstacle in the application of deep learning to medical imaging due to the challenges associated with collecting and annotating a wide range of medical images. This scarcity can result in issues such as overfitting and biased outcomes in the models. To tackle this issue, methods for data augmentation are utilized to automatically boost the diversity and volume of training datasets[38].In our case, we were dealing with two limited datasets, one as a baseline and the other as a target, both contains limited number of 15 and 16 patients. Nevertheless, it is still not clear which augmentation techniques are most effective for specific types of medical images, as their efficacy varies based on the disease and imaging method used.

In this work, we incorporated various image augmentation approaches to increase the diversity of medical imaging datasets. The following provides an in-depth explanation of each augmentation method employed in the ”augment” function:

3.4.2 No Augmentation

The original image remains unaltered, serving as a reference point for evaluating the effects of other augmentation techniques.

3.4.3 Vertical Translation Downwards

This approach moves the image downward vertically by a specified amount. The upper section of the image is filled with a background value based on the image type, simulating a downward movement of the scanning device or subject.

3.4.4 Vertical Translation Upwards

The image is shifted upward vertically by a given amount, with the bottom part of the image filled with an appropriate background value. This simulates an upward movement of the scanning device or subject.

3.4.5 Horizontal Translation Rightwards

This technique shifts the image horizontally to the right by a certain amount. The left side of the image is filled with a background value suitable for the image type, creating an effect similar to the subject or scanning device moving to the right.

3.4.6 Horizontal Translation Leftwards

The image is shifted horizontally to the left by a specific amount, with the right side filled with an appropriate background value. This effect is similar to the subject or scanning device moving to the left.

3.4.7 Horizontal Flip

The image is horizontally mirrored, reflecting it across its vertical axis. This technique is particularly valuable for addressing asymmetries in the dataset.

3.4.8 Vertical Flip

The image is vertically mirrored, flipped upside down, and reflected across its horizontal axis. This method can help the model generalize across different orientations.

3.4.9 Horizontal and Vertical Flip

This approach combines both horizontal and vertical flips, resulting in an image rotated by 180 degrees. This augmentation is used to create a more comprehensive dataset that includes all possible orientations. These augmentation methods were thoughtfully chosen to improve the model's capacity to generalize across variations in medical imaging, ultimately enhancing the accuracy and resilience of diagnostic outcomes. Each augmentation type is designed to simulate different real-world scenarios or potential inconsistencies in image acquisition.

3.5 Model

3.5.1 Model Architecture

The 2D U-net variant used in this study was proposed by Spadea et al[2] and is illustrated in Fig 3.7. This model follows a U-Net style network structure with separate encoding and decoding paths and aims to convert input T1-weighted Magnetic Resonance (T1MR) images into synthetic Computed Tomography (sCT) slices.

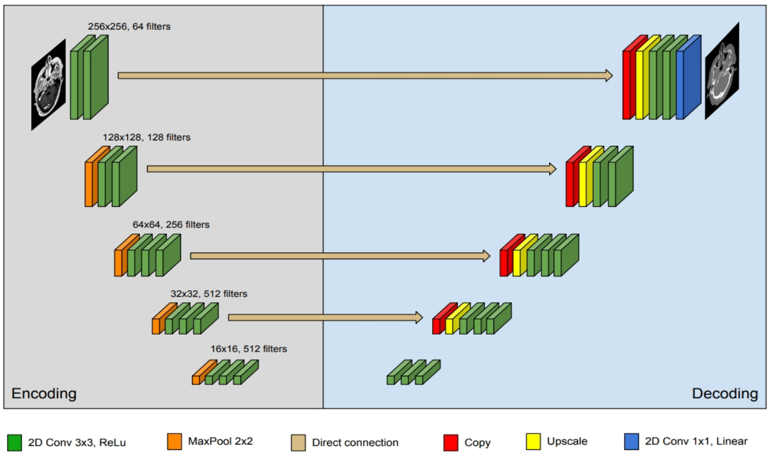


Figure 3.7: 2D Unet Model Architecture

The deep learning model was making use of open-source PyTorch packages and incorporating parameters from Spadea et al [2] . The main purpose of our architectural modification is for educational exploration. The network is structured into five levels and is divided into two symmetrical sections: the encoding branch and the decoding branch.

The architecture starts with an input layer where the initial number of feature maps and MRI channels are defined using a dictionary, **‘lp‘** which is the learning parameters file. The model then moves through a contracting path, which includes various convolutional blocks with dropout layers to prevent overfitting.

In the contracting path, the initial layer, *‘SingleConv3x3‘*, processes the input channels and generates initial feature maps based on the parameters specified in ‘lp‘. This is followed by another *‘SingleConv3x3‘* layer with the same number of feature maps, following a *‘2x2 MaxPooling‘* layer then reducing the spatial dimensions by half.

After these layers, a *‘DoubleConv3x3‘* layer is used to double the number of feature maps, which is then followed by another *MaxPooling* layer that further reduces the feature map dimensions. The path continues with a *‘TripleConv3x3‘* layer, which quadruples the number of features compared to the initial layers.

As the contracting path progresses, another *MaxPooling* layer is used to reduce the spatial dimensions again. This pattern follows the U-Net architecture, gradually decreasing the spatial size while increasing depth until the bottleneck is reached.

The expansive path starts at this point and mirrors the contracting path but in reverse. It includes an upsampling sequence with *‘UpConcat‘* layers, utilizing *‘nn.ConvTranspose2d‘* and bilinear upsampling to restore spatial dimensions. The first *‘UpConcat‘* layer concatenates the output from the bottleneck with the corresponding feature map from the contracting path, following the U-Net paradigm of skip connections.

In the extended pathway, the combined feature maps undergo processing via another *'TripleConv3x3'* layer to enhance the features. Following this, the pathway includes additional *'Up-Concat'* and convolutional layers to progressively restore the spatial dimensions and merge features from the corresponding contracting path layers.

Upon completion of the extended pathway, a *'SingleConv1x1'* layer is utilized by the model to transform the final feature maps to the necessary number of output classes. This ultimate convolutional layer consolidates the features into a two-dimensional output.

Dropout layers are strategically positioned throughout the architecture, occurring after pooling operations and at crucial points in the extended pathway to prevent overfitting while retaining essential feature information. This approach ensures robust feature extraction and the effective integration of multiscale contextual information, rendering it suitable for research purposes, although it is not specifically optimized for clinical use.

The network is trained using mean absolute error as the cost function and incorporates L1 regularization to counteract overfitting.

Block	**Layers**	**Dropout**
Down 1	Conv2D+ReLU+BNMaxPool+Dropout	Yes (after MaxPool)
Down 2	Conv2D+ReLU+BNMaxPool+Dropout	Yes (after MaxPool)
Down 3	Conv2D+ReLU+BNMaxPool+Dropout	Yes (after MaxPool)
Down 4	Conv2D+ReLU+BNMaxPool+Dropout	Yes (after MaxPool)
Bottleneck	Conv2D+ReLU+BNConv2D+ReLU+BN	No
Up 1	Conv2D+ReLU+BNConvTransp+Dropout	Yes (after ConvTransp)
Up 2	Conv2D+ReLU+BNConvTransp+Dropout	Yes (after ConvTransp)
Up 3	Conv2D+ReLU+BNConvTransp+Dropout	Yes (after ConvTransp)
Output	Conv2D+ReLU+BN	Yes (before final Conv2D)

Table 3.1: Description of the 2D U-Net Model Architecture

3.6 TRAINING AND TESITNG

3.6.1 TRAINING ON THE BASELINE DATASET

In this stage, the model is trained on the baseline dataset using the following **hyperparameters**:

- **Learning Rate**: 5.0×10^{-5}

- **Epochs**: 80
- **Lambda**: 4.0×10^{-4}
- **Batch Size**: 20
- **Loss Function**: Masked
- **Initial Number of Filters**: 64
- **Dropout Rate**: 1.0×10^{-1}

The primary objective during this phase is to establish a robust foundation by training the model from scratch. The learning rate of 5.0×10^{-5} is selected to ensure that the model updates weights gradually, preventing large oscillations in the loss function, which is particularly important when working with complex medical images. The model is trained for 80 epochs, which strikes a balance between sufficient training time and the risk of overfitting.

A key aspect of this phase is the use of the **masked loss function**, which helps focus the training process on regions of interest within the images, effectively ignoring irrelevant areas that might introduce noise into the learning process.

Additionally, the training process incorporates a **lambda regularization term (set at 4.0×10^{-4})** to prevent overfitting by penalizing overly complex models. The **batch size** of 20 is chosen to provide a reasonable trade-off between memory efficiency and the ability to update model weights frequently. The **dropout rate** of 1.0×10^{-1} is introduced as a regularization technique to further mitigate overfitting by randomly deactivating a fraction of neurons during each iteration.

During this phase, the model's weights are saved whenever an improvement in performance is observed. This ensures that the best-performing model at any given epoch is preserved, providing a solid starting point for the next phase of training.

3.6.2 TRAINING ON THE TARGET DATASET

After establishing a well-trained model on the baseline dataset, the next step is to transfer the learned knowledge to the target dataset. This process is carried out through **transfer learning**, a powerful technique that leverages pre-trained models on a new, often smaller, dataset.

Transfer learning is particularly effective in scenarios where the target dataset is limited in size or has a different distribution compared to the baseline dataset. By initializing the model with weights learned from the baseline dataset, we can significantly reduce the training time and improve model performance on the target dataset. This approach is rooted in the idea that early layers of a neural network often learn general features, such as edges and textures, which are applicable across different datasets, while later layers learn more specific features relevant to the task at hand[39].

In this case, the model's weights, saved from the baseline training phase, are used as the starting point for training on the target dataset. However, instead of training all layers from scratch, we **freeze the early convolutional layers (Convdown 1 and 2)**. This means that the weights in these layers are not updated during training on the target dataset. Freezing these layers allows the model to retain the general features learned during the baseline training, while focusing on fine-tuning the later layers to adapt to the specific characteristics of the target dataset. Those changed **hyperparameters** are:

- **Learning Rate**: 1.25×10^{-5}
- **Dropout Rate**: 1.0×10^{-1}
- **Batch Size**: 5
- **Frozen Layers**: *Convdown 1, Convdown 2*

The **learning rate** is further reduced to 1.25×10^{-5} to allow for slower, more careful updates to the model weights, ensuring that the model fine-tunes effectively to the new data without disrupting the learned general features. The **dropout rate** remains at $1.0 \times 10^{-1}$to maintain regularization, while the **batch size** is reduced to 5. The reduction in batch size is based on experimental observations that indicated better results with slower convergence, allowing the model to reach a smaller loss.

By freezing the early layers and using a smaller batch size, the training process achieves a more refined adaptation to the target dataset. This method helps avoid overfitting while making efficient use of the knowledge obtained during the baseline training phase.

3.6.3 LEAVE-ONE-OUT CROSS-VALIDATION

The Baseline dataset was divided into training, validation, and testing sets with 12, 2, and 1 patient(s) respectively. By the same token we divided the target dataset into 12, 3, and 1 patient(s), since we have one patient more. To enhance the size of the testing dataset, a leave-one-out cross-validation approach was applied to each of the 15 patients.

Leave-One-Out Cross-Validation (LOOCV) is a method used to evaluate machine learning algorithms. This method entails training a model using all data points except one and then testing the model on the single omitted data point. Each data point in the dataset undergoes this process repeatedly.LOOCV offers a nearly unbiased estimate of model performance, making it useful for small datasets.Nonetheless, it can be computationally intensive due to the substantial number of training iterations needed[40]. In this method, one patient was repeatedly selected as the testing data, while the remaining 14 patients were utilized for training and validating the network.

3.6.4 TRAINING ELEMENTS

In the training procedure, we utilize a masked mean absolute error **(MAE)** loss function, concentrating on critical image regions to calculate errors accurately.To ensure stable parameter updates, we use Stochastic Gradient Descent **(SGD)** with momentum, and we prevent overfitting by penalizing large weights with weight decay **(lambda)**.We start with a low learning rate of 5×10^{-5} over 80 epochs to facilitate gradual and precise learning.We chose a batch size of 18 based on our computational resources. Furthermore, we apply a dropout rate of 1.0×10^{-1}, which helps in reducing overfitting by randomly excluding neurons during each training iteration, thus improving the model's generalization.

We train each view independently, meaning we separately train 2D CNNs for axial, sagittal, and coronal views of MRI images. This method allows each CNN to focus on and optimize learning for its specific orientation, accommodating the unique characteristics and variations within each plane. As a result, each CNN has the potential to achieve more accurate convergence specific to its view.

In the next step we should combine generated slices from all views. The voter plays a crucial role in combining the outputs of these independently trained CNNs by utilizing a median voting strategy to integrate the predictions from the axial, sagittal, and coronal CNNs. This strategy involves taking the median value of the predictions for each corresponding pixel across all three planes. By doing so, it effectively reduces the impact of outlier predictions from any single plane and leverages the strengths of each plane's perspective. Therefore, the voter produces a more robust and accurate sCT prediction by synthesizing the diverse insights provided by the three different orientations.

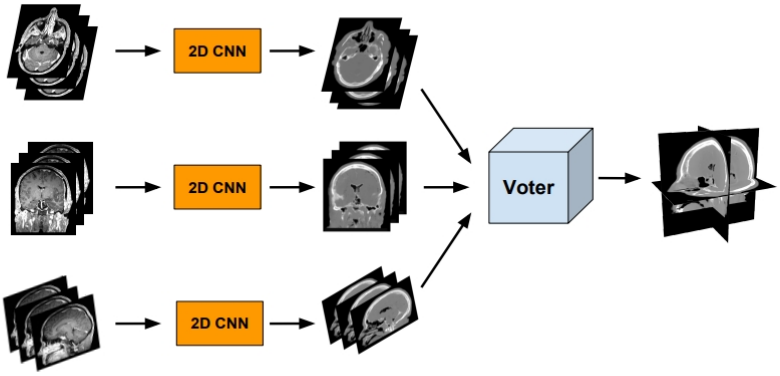


Figure 3.8: combining slices from independent training [2]

3.6.5 TESTING

The process of testing includes the comparison of the synthetic CT images' predictions with the actual CT images within a defined region of interest determined by a mask[2]. Two metrics are used for evaluation:

MAE: Represents the average absolute variance between the HU of the predicted and real CT values, indicating the overall precision.

$$\text{MAE} = \frac{1}{M} \sum_{i=1}^{M} |sHU_i - HU_i| \tag{3.1}$$

where sHU and HU are the Hounsfield Unit values measured on sCT and CT, respectively.

BIAS: Represents the Mean Error (ME) error, showing whether the predictions consistently overestimate or underestimate the actual CT values

$$\text{BIAS} = \frac{1}{M} \sum_{i=1}^{M} (sHU_i - HU_i) \tag{3.2}$$

The **DSC**, or **Dice similarity coefficient**, is another widely used measure for evaluating the precision of representing specific tissue classes or structures such as bones, fat, muscle, air, and the body. In this scenario, the DSC is computed after applying a threshold to CT and sCT, and performing morphological operations on the binary masks if necessary[41].

$$\text{DSC} = \frac{2|A \cap B|}{|A| + |B|}$$

where A and B represent the binary masks of the segmented structures.

Furthermore, metrics commonly employed for assessing segmentation accuracy, such as Hausdorff distance or mean absolute surface distance, can also be utilized. These metrics calculate the maximum and average distance between two sets of contours.

4

Results

4.1 PREPROCESSING

4.1.1 DATA QUALITY

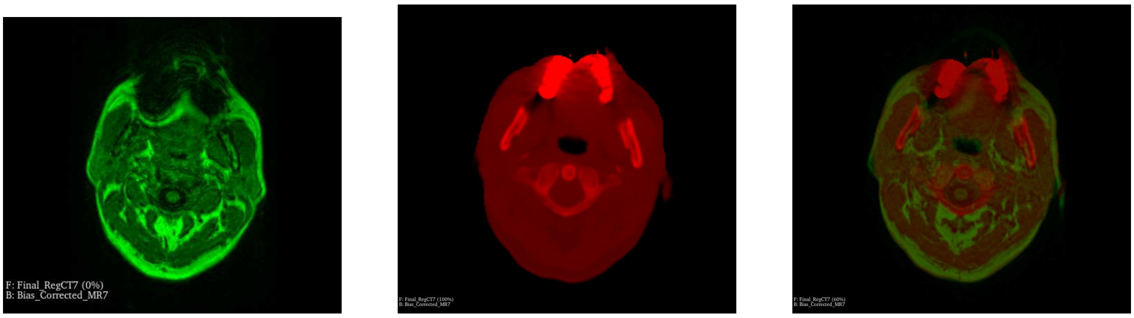
Quality data plays a crucial role in the analysis of medical imaging, particularly when making use of deep learning models. The effectiveness of these models depends on having large, accurately labeled datasets, which require significant expertise and time to create. Poor quality data or excessive labeling can harm model performance[42].

Data quality plays a significant role also in optimal masking and final registration in our preprocessing procedure. We are dealing with a target dataset that comes from neurodegenerative patients, and generally, we are dealing with some challenges which are :

4.1.2 METAL-INDUCED SUSCEPTIBILITY ARTIFACTS

The presence of metallic implants in MRI can cause substantial image artifacts, including signal loss, failure of fat suppression, geometric distortion, and bright pile-up artifacts. These cause large resonant frequency changes and failure of many MRI mechanisms. Artifacts can be minimized or avoided through careful selection of parameters and pulse sequences, although more advanced imaging techniques can provide additional improvements in imaging.[43]. We have two patients(P7, P18) with MRI scans the metal artifacts made the whole mouth blurry, causing accurate rigid registration challenging because the mask from the MRI is not fit to make from the CT. The level of blurriness in P18 was so adverse that we decided to exclude it from the training procedure.

In Figure 4.1a the green image shows a T1 MR scan with a noticeable cavity caused by the artifact. The red image is the corresponding CT scan after rigid registration, where the cavity from the MR mask is also visible. On the right, you can see an overlay of the two images



(a) Case7 T1 Axial view

(b) Case7 Registered CT Axial view

(c) Overlay of RegCT and T1

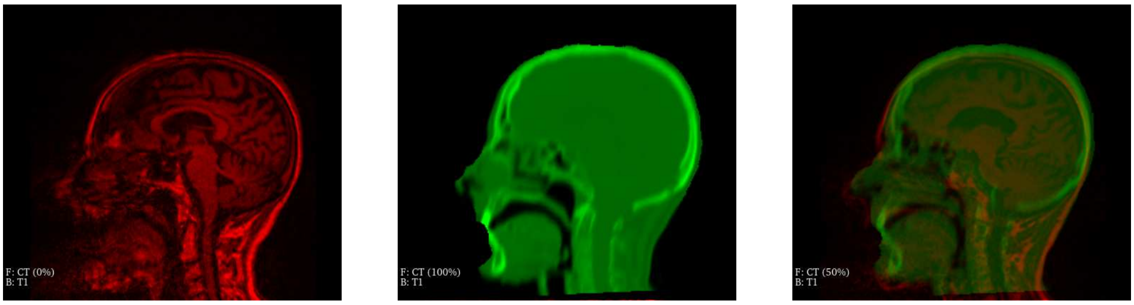
Figure 4.1: Metal artifacts

4.1.3 MOTION ARTIFACTS

Motion artifacts have posed significant challenges in MRI since its inception, particularly in neurodegenerative patients who often experience involuntary movements.These artifacts, including blurring and ghosting, are commonly seen in clinical imaging and can reduce image quality and impede accurate diagnosis. Despite more than thirty years of study and the development of many methods to reduce or fix motion artifacts, there is no single solution that works universally[44].

Our dataset contains neurodegenerative cases so involuntary movements during scanning are a unique characteristic of these patients. We observed noises induced by motion artifacts in two cases significantly (P4, P22).

In Figure 4.2a the red image displays a T1 MR scan with noticeable motion artifacts. The green image is the corresponding CT scan after rigid registration, where the MR mask was not precisely aligned. On the right, an overlay of the two images highlights these discrepancies.



(a) Case4 T1 sagittal view

(b) Case4 Registered CT sagittal view

(c) Overlay of RegCT and T1

Figure 4.2: Motion artifacts

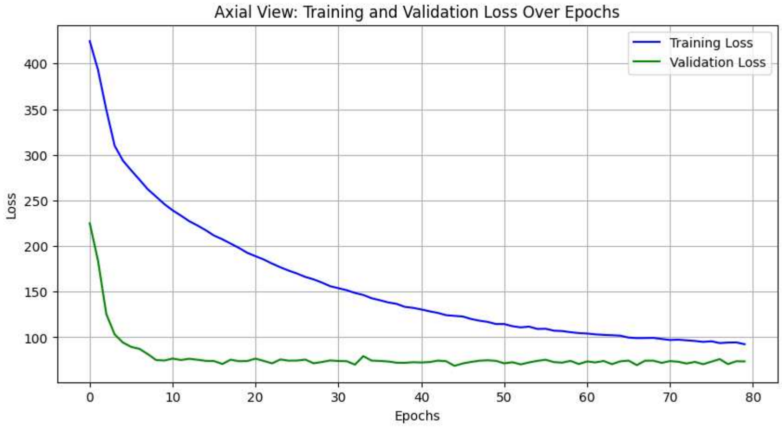
4.1.4 Low-artifact cases facilitating optimal preprocessing

Here we deal with fewer artifacts and noises, giving us high-quality data for further training and analysis. These cases made the whole process of masking and registration straightforward since adjusting the mask wasn't needed manually. some examples are: P3,P9,P15 and P19. The rest of the cases still got some artifacts but, they weren't as challenging as the two first categories with artifacts that we discussed previously.

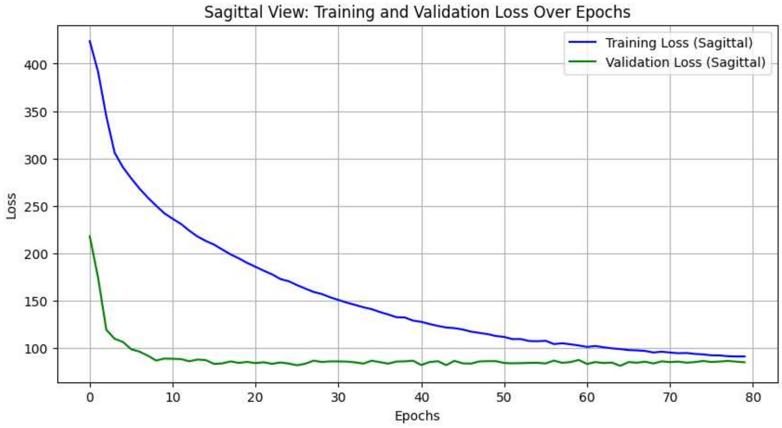
Figure 4.3 is an example case (P3) with Minimal Artifacts: The blue CT image, manually registered, is shown over the red MR image in the background, viewed from the sagittal plane. This alignment demonstrates the quality of the registration with minimal artifacts

Figure 4.3: Example Case (P3) with Minimal Artifacts

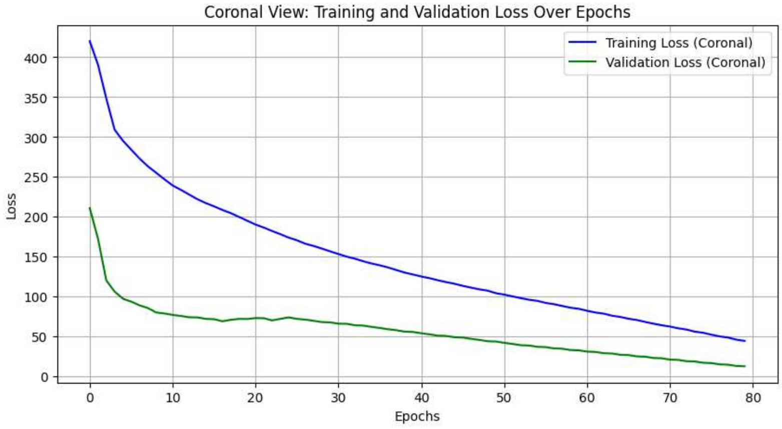
4.2 BASELINE DATASET



(a) Axial View



(b) Sagittal View



(c) Coronal View

Figure 4.4: Training and validation loss for the three views: axial, sagittal, and coronal.

The dataset was previously utilized by Spadea et al.[2], and the sCT analysis has been thoroughly presented in their work. We conducted only one training session, as our primary objective was to save the trained weights for initialization in our target dataset if our baseline model demonstrated significant improvement in both training and validation loss. In our testing case (Patient 23), the model produced a mean absolute error (MAE) of 60.437 and a bias (mean error, ME) of 19.909. The figure 4.4 illustrate the training and validation loss across all three views: axial, sagittal, and coronal.

4.3 TARGET DATASET

4.3.1 QUANTITATIVE ERROR METRICS

The quantitative evaluation of the sCT images was further detailed in a table of MAE and ME values across all 15 patients. The Mean Absolute Error (MAE) gives the average absolute value of errors, disregarding their direction. Meanwhile, the Mean Error (ME) provides the average signed error, indicating whether the sCT tends to overestimate or underestimate the true CT values. The trends observed in these metrics across patients align with the visual and DSC analysis, confirming that image noise and patient-specific factors significantly impact the accuracy of sCT reconstruction.

Table 4.1: The Quantitative Evaluation of the sCT Images

Patient	**MAE (HU)**	**ME**
Patient 3	75.4	−12.2
Patient 4	196.5	78.4
Patient 5	126.5	40.0
Patient 7	101.0	49.6
Patient 8	96.5	15.7
Patient 9	77.5	26.4
Patient 10	74.7	31.7
Patient 11	106.1	35.4
Patient 12	123.0	48.0
Patient 15	76.2	5.1
Patient 16	122.8	46.1
Patient 17	58.9	30.0
Patient 19	81.3	20.0
Patient 20	117.5	54.0
Patient 22	139.3	31.1
Mean	**104.88**	**33.29**

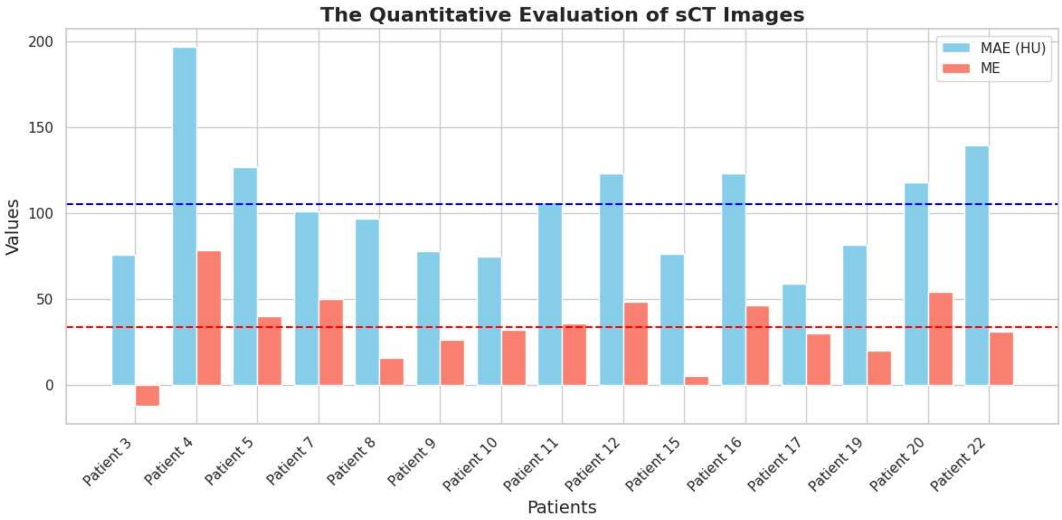


Figure 4.5: MAE and ME for all cases

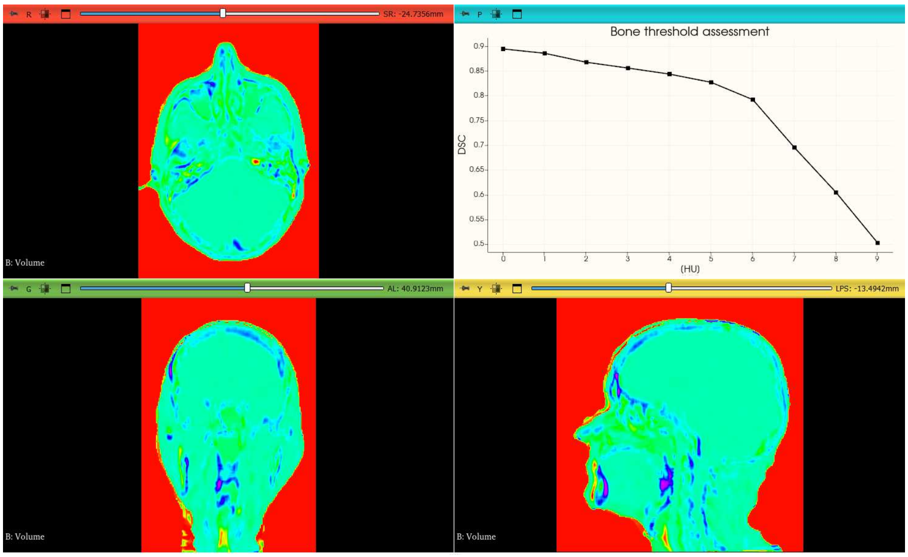
4.3.2 sCT and CT Output Error Maps

In the analysis of the synthetic CT (sCT) images versus the CT images, output error maps were generated using 3D Slicer for three anatomical views: sagittal, axial, and coronal. These maps illustrate the regions of discrepancies between the sCT and the actual CT images. The visualization is color-coded to reflect the magnitude of error, where green and blue indicate lower discrepancies, and red areas signify higher discrepancies. The quality of the initial scan was found to have a strong correlation with the accuracy of the resulting sCT. In particular, scans with more noise, which is often a result of poor image quality, were linked to higher Mean Absolute Error (MAE) and Mean Error (ME) values in the sCT images.This finding is consistent across the sample dataset, which primarily includes older patients suffering from neurodegenerative diseases. Such conditions typically contribute to image artifacts and noise, exacerbating the error between sCT and actual CT, as illustrated by the output error maps.

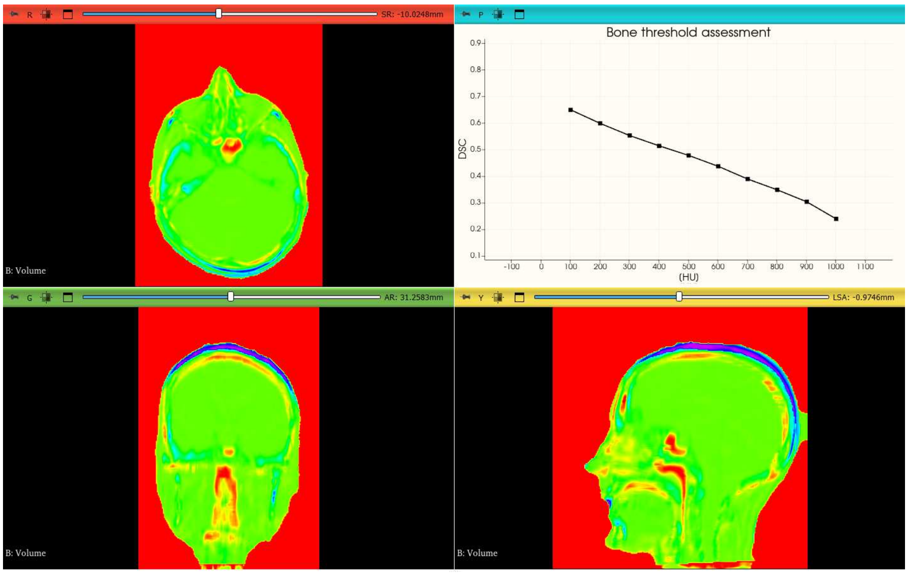
4.3.3 Bone Threshold Assessment

The accuracy of the sCT images was further evaluated by conducting a bone threshold assessment. The diagram on the right side of the color maps illustrates the analysis, showing a plot of the Dice Similarity Coefficient (DSC) against different Hounsfield Unit (HU) thresholds. The DSC is a measurement used to assess the overlap between the segmented bone structures in the sCT and real CT images, with a score of 1 denoting perfect overlap and 0 indicating no overlap.

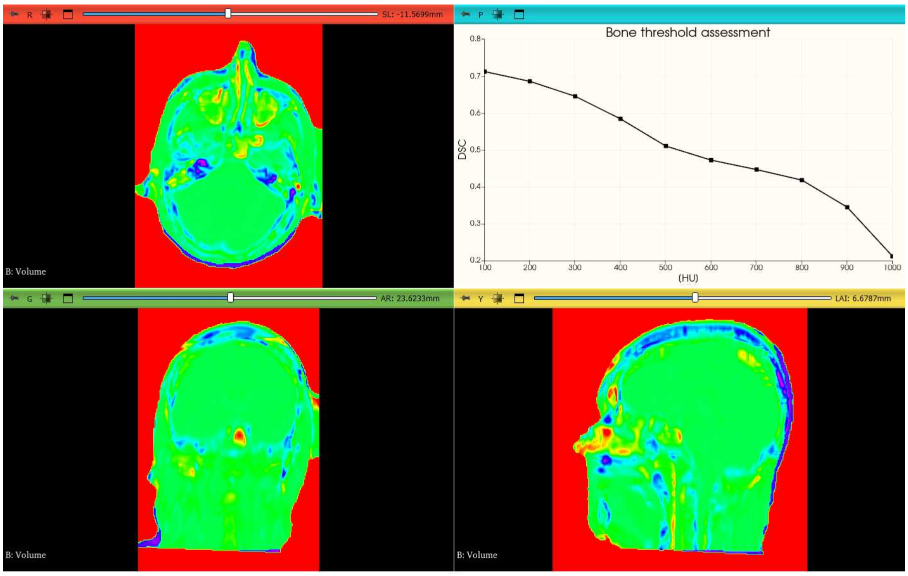
The diagram reveals a trend where the DSC values decrease as the HU threshold increases. Initially, at lower HU thresholds, the DSC is higher, suggesting that the sCT accurately captures the bone structures. However, as the threshold increases, the DSC drops, indicating that higher-density bone tissues are not as well-represented in the sCT images compared to the actual CT images. This decline in DSC with increasing HU is expected as the higher thresholds correspond to denser bone tissues, which are more challenging to accurately model in synthetic imaging.



(a) Patient 3

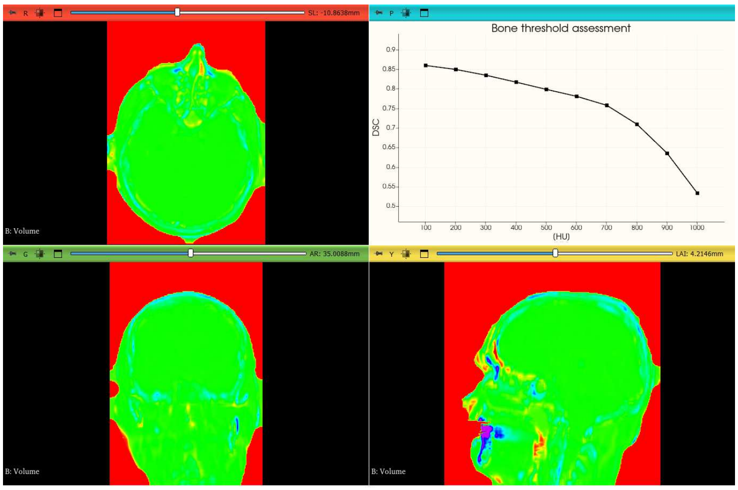


(b) Patient 4

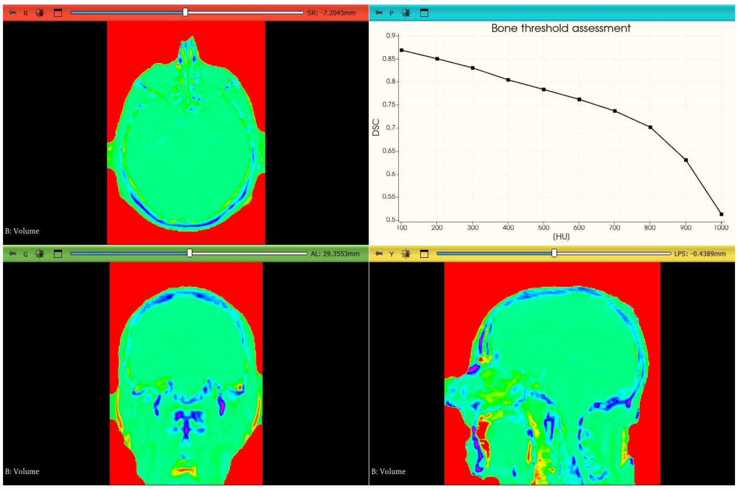


(c) Patient 5

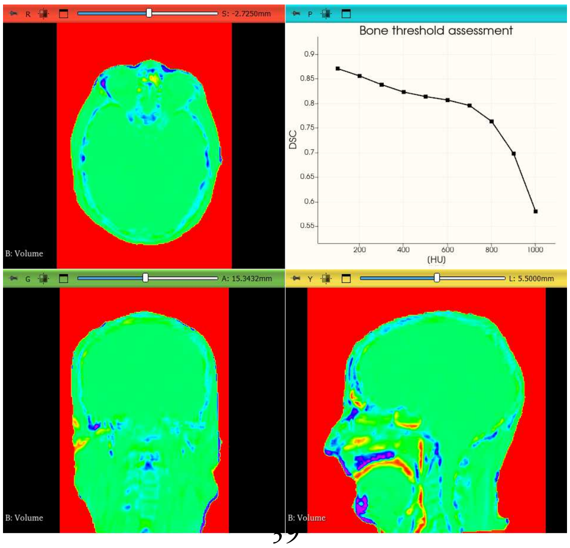
Figure 4.6: Output error maps for Patients 3-5 in sagittal, axial, and coronal views.



(a) Patient 7

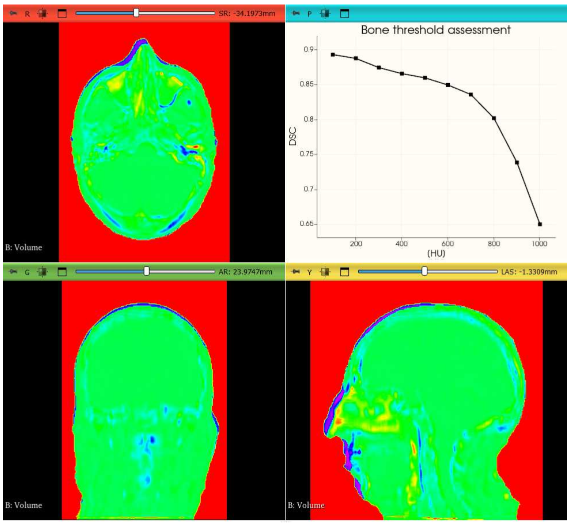


(b) Patient 8

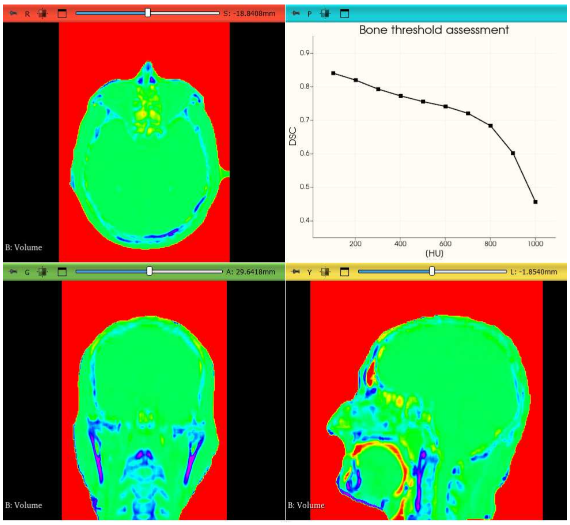


(c) Patient 9

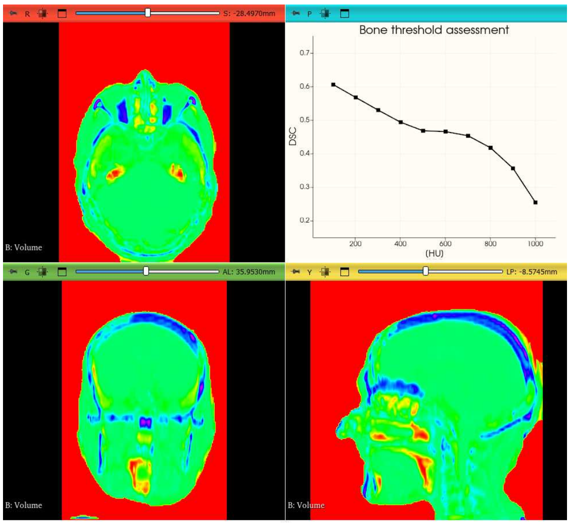
Figure 4.7: Output error maps for Patients 7-9 in sagittal, axial, and coronal views.



(a) Patient 10

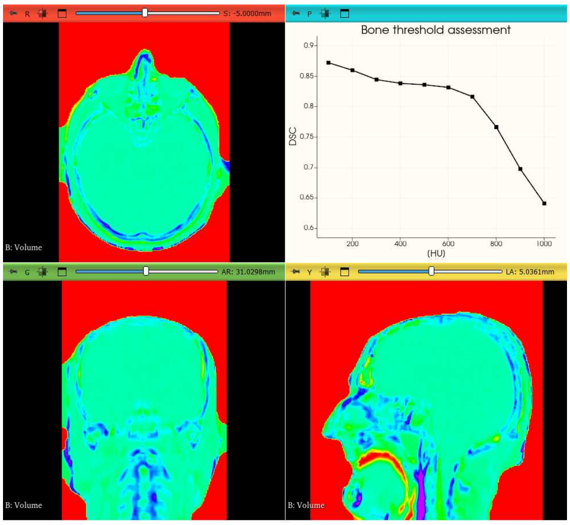


(b) Patient 11

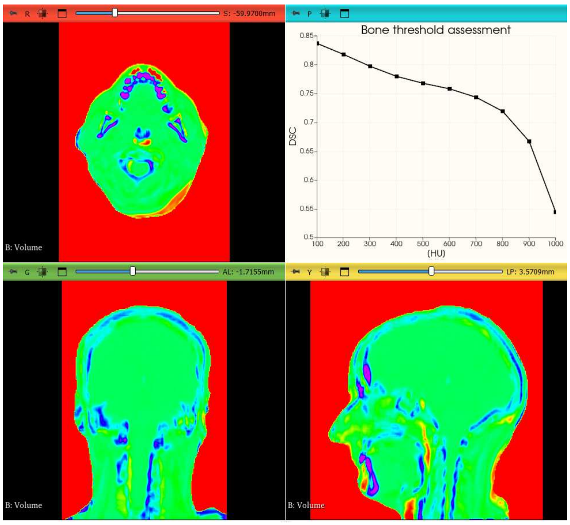


(c) Patient 12

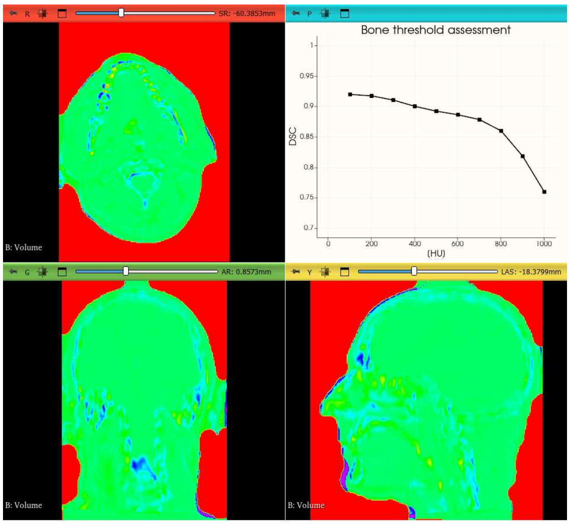
Figure 4.8: Output error maps for Patients 10-12 in sagittal, axial, and coronal views.



(a) Patient 15

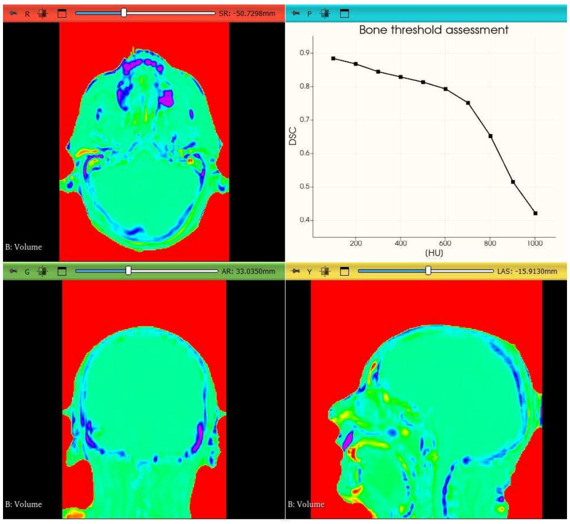


(b) Patient 16

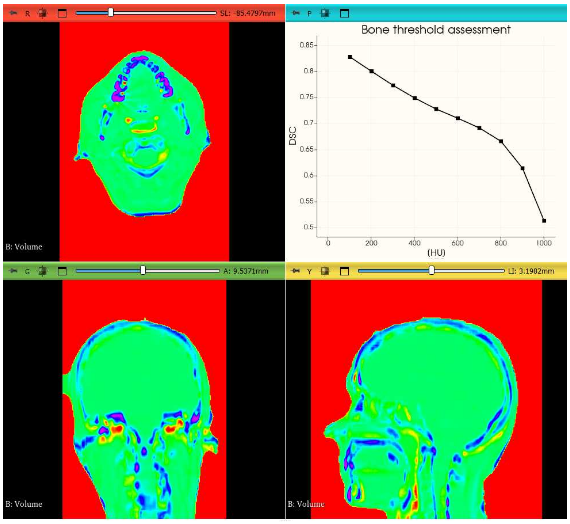


(c) Patient 17

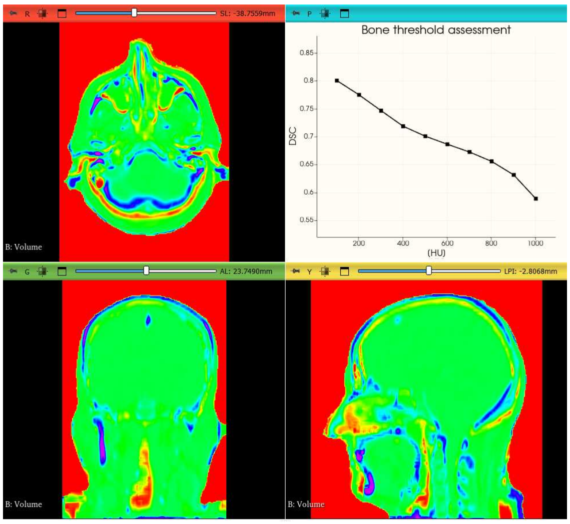
Figure 4.9: Output error maps for Patients 13-15 in sagittal, axial, and coronal views.



(a) Patient 19



(b) Patient 20



(c) Patient 22

Figure 4.10: Output error maps for Patients 19, 20, and 22 in sagittal, axial, and coronal views.

5

Conclusion

This thesis explored the application of a 2D U-Net deep learning model to generate synthetic CT (sCT) images from MRI scans, aiming to reduce the need for multiple CT scans in radiation therapy planning. The study, which focused on neurodegenerative patients, demonstrated that the U-Net model, combined with a comprehensive preprocessing pipeline and transfer learning, can produce sCT images that closely approximate the ground truth CT images. However, the model's performance was affected by the quality of the input MRI data, particularly in cases with significant artifacts.

The generation of sCT images from MR or CBCT scans has the potential to significantly improve the radiotherapy process by reducing the patient's exposure to ionizing radiation and increasing the precision of the treatment. Currently, deep learning algorithms are the most effective techniques for image translation. However, to ensure the safe and reliable clinical use of sCTs, it is crucial to develop an independent system that can verify the accuracy of each synthetic CT generated during the treatment workflow. Even minor variations in intensity, invisible to the naked eye, can potentially compromise treatment outcomes.

At present, there are no established methodologies for quality assurance in this domain. Future research should focus on designing, developing, and validating methods to identify synthetic CTs that contain translation errors undetectable by human observation. By intercepting these errors, we can prevent incorrect images from being used in patient treatment. Additionally, by calculating translation errors, this information could be leveraged to further refine and improve current conversion algorithms.

In conclusion, while this study contributes to the development of synthetic CT in radiotherapy planning, ensuring its safe clinical implementation will require significant advancements in quality assurance methods. These efforts will be essential to fully realize the potential of synthetic CTs in improving patient outcomes in radiotherapy.

References

[1] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[2] M. F. Spadea, G. Pileggi, P. Zaffino, P. Salome, C. Catana, D. Izquierdo-Garcia, F. Amato, and J. Seco, “Deep convolution neural network (dcnn) multiplane approach to synthetic ct generation from mr images: Application in brain proton therapy,” *International Journal of Radiation Oncology*, vol. 10.1016/j.ijrobp.2019.04.044, 2019.

[3] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, “Deep learning for healthcare: Review, opportunities and challenges,” *Briefings in Bioinformatics*, vol. 10.1093/bib/bbx044, 2018.

[4] T. Wang, Y. Lei, Y. Fu, J. F. Wynne, W. J. Curran, T. Liu, and X. Yang, “A review on medical imaging synthesis using deep learning and its clinical applications,” *Journal of Applied Clinical Medical Physics*, vol. 10.1002/acm2.13121, pp. 11–36, 2021.

[5] Z. Eidex, Y. Ding, J. Wang, E. Abouei, R. L. J. Qiu, T. Liu, T. Wang, and X. Yang, “Deep learning in mri-guided radiation therapy: A systematic review,” *Journal of Applied Clinical Medical Physics*, vol. 10.1002/acm2.14155, 2023.

[6] S. E. Whang, Y. Roh, H. Song, and J.-G. Lee, “Data collection and quality challenges in deep learning: A data-centric ai perspective,” *Journal Name*, vol. 32, pp. 791–813, 2023.

[7] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” *International Conference on Medical Image Computing and Computer-Assisted Intervention*, vol. 10.1007/978-3-319-24574-4$_2$8$, pp.234 - -241, 2015.$

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 10.1109/CVPR.2016.90, pp. 770–778, 2015.

[9] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 10.1109/CVPR.2018.00745, pp. 7132–7141, 2017.

[10] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 10.1109/CVPR.2017.243, pp. 4700–4708, 2017.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 10.1145/3422622, pp. 139–144, 2020.

[12] H. Emami, M. Dong, Y. Ding, and C. Glide-Hurst, "Magnetic resonance imaging (mri)-only radiation therapy: Synthetic computed tomography (sct) generation using gans," *International Journal of Radiation Oncology*Biology*Physics*, vol. 10.1016/j.ijrobp.2018.03.001, pp. 806–812, 2018.

[13] J. M. Wolterink, A. M. Dinkla, M. H. F. Savenije, P. R. Seevinck, C. A. T. van den Berg, and I. Išgum, "Deep mr to ct synthesis using unpaired data," *IEEE International Symposium on Biomedical Imaging (ISBI)*, vol. 10.1109/ISBI.2017.7950667, pp. 225–228, 2017.

[14] S. Kazemifar, S. McGuire, R. Timmerman, Z. Wardak, D. Nguyen, Y. Park, S. Jiang, and K. Sheng, "Dosimetric evaluation of synthetic ct generated with gans for mri-only brain radiotherapy," *Radiotherapy and Oncology*, vol. 10.1016/j.radonc.2020.04.012, pp. 240–248, 2020.

[15] E. Gong, J. M. Pauly, and G. Zaharchuk, "Cycle-consistent generative adversarial networks for mr-ct translation in head and neck cancer patients," *Journal of Magnetic Resonance Imaging*, vol. 10.1002/jmri.26860, pp. 1244–1253, 2019.

[16] J. Xue, Y. Liu, J. Wang, W. Li, L. Du, H. Liu, and G. Li, "Unpaired mr-to-ct synthesis using a cycle-consistent generative adversarial network with adversarial and perceptual losses," *IEEE Access*, vol. 10.1109/ACCESS.2021.3058821, pp. 33 585–33 593, 2021.

[17] C. Wang, X. Zhu, J. C. Hong, and D. Zheng, "Artificial intelligence in radiotherapy treatment planning: Present and future," *Technology in Cancer Research Treatment*, vol. 10.1177/1533033819873922, 2019.

[18] S. Devic, "Mri simulation for radiotherapy treatment planning," *Medical Physics*, vol. 10.1118/1.4758068, pp. 6701–6711, 2012.

[19] M. Maspero, P. R. Seevinck, G. Schubert, and C. A. T. van den Berg, "Synthetic ct in mri-only radiation therapy: A critical review," *Radiotherapy and Oncology*, vol. 10.1016/j.radonc.2020.03.006, pp. 147–154, 2020.

[20] A. Mehranian and H. Zaidi, "Attenuation correction for pet/mri: Challenges and opportunities," *European Journal of Nuclear Medicine and Molecular Imaging*, vol. 10.1007/s00259-014-2969-1, pp. 681–694, 2015.

[21] A. Martinez-Möller and S. G. Nekolla, "A review of attenuation correction techniques for pet/mri," *Journal of Nuclear Medicine*, vol. 10.2967/jnumed.111.096321, pp. 59–63, 2012.

[22] S. Avanzo, Wei, "Artificial intelligence and machine learning in precision oncology: A review on enhancing discoverability through multiomics integration," *British Journal of Radiology*, p. e185–e202, 2024.

[23] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, pp. 386–408, 1958.

[24] C. Zuo, J. Qian, S. Feng, W. Yin, Y. Li, P. Fan, J. Han, K. Qian, and Q. Chen, "Deep learning in optical metrology: A review," *Light: Science Applications*, vol. 10.1038/s41377-022-00738-x, 2022.

[25] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, ser. Adaptive Computation and Machine Learning Series. Cambridge, Massachusetts; London, England: MIT Press, 2016.

[26] I. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *SN Computer Science*, vol. 10.1007/s42979-021-00592-x, p. 160, 2021.

[27] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, pp. 193–202, 1980.

[28] IBM. Convolutional neural networks (cnns): Overview, applications, and examples. [Online]. Available: https://www.ibm.com/topics/convolutional-neural-networks

[29] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: An overview and application in radiology," *Insights into Imaging*, vol. 9, pp. 611–629, 2018.

[30] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.

[31] ML4A. How neural networks are trained. [Online]. Available: https://ml4a.github.io/ml4a/how_neural_networks_are_trained/

[32] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, 2019.

[33] A. Ng, *Machine Learning Yearning*. Self-published, 2017.

[34] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," *arXiv preprint*, vol. 1206.5533, 2012.

[35] Susanne Schmid. Image intensity normalization in medical imaging. [Online]. Available: https://medium.com/@susanne.schmid/image-normalization-in-medical-imaging-f586c8526bd1

[36] S. Song, Y. Zheng, and Y. He, "A review of methods for bias correction in medical images," *Biomedical Engineering Review*, vol. 1, 2017.

[37] P. Paavilainen, S. U. Akram, and J. Kannala, "Bridging the gap between paired and unpaired medical image translation," *Electrical Engineering and Systems Science > Image and Video Processing*, vol. 1, 2021.

[38] E. Goceri, "Medical image data augmentation: Techniques, comparisons and interpretations," *Artificial Intelligence Review*, vol. 10.1007/s10462-023-10453-z, pp. 12 561–12 605, 2023.

[39] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," *International Conference on Artificial Neural Networks*, pp. 270–279, 2018.

[40] J. Brownlee. Loocv for evaluating machine learning algorithms. [Online]. Available: https://machinelearningmastery.com/loocv-for-evaluating-machine-learning-algorithms/

[41] M. F. Spadea, M. Maspero, P. Zaffino, and J. Seco, "Deep learning based synthetic-ct generation in radiotherapy and pet: A review," *Journal Name*, vol. DOI or volume number, p. Page range (if available), 2023.

[42] A. R. Luca, T. F. Ursuleanu, L. Gheorghe, R. Grigorovici, S. Iancu, M. Hlusneac, and A. Grigorovici, "Impact of quality, type and volume of data used by deep learning models in the analysis of medical images," *Informatics in Medicine Unlocked*, vol. 10.1016/j.imu.2022.100911, 2022.

[43] B. A. Hargreaves, P. W. Worters, K. Butts Pauly, J. M. Pauly, K. M. Koch, and G. E. Gold, "Metal-induced artifacts in mri," *American Journal of Roentgenology*, vol. 10.2214/AJR.11.7364, 2011.

[44] M. Zaitsev, J. Maclaren, and M. Herbst, "Motion artifacts in mri: A complex problem with many partial solutions," *Journal of Magnetic Resonance Imaging*, vol. 10.1002/jmri.24850, 2015.

Acknowledgments

The author wishes to express sincere gratitude to the Karlsruhe Institute of Technology, Institut für Biomedizinische Technik (IBT), for providing the infrastructure and support necessary for this research. Special thanks to Prof. Dr.-Ing. Maria Francesca Spadea, whose guidance was highly influential throughout this research, as well as to Prof. Paolo Zaffino and MSc. Ciro Benito Raggio for their valuable insights and direction.

The author also extends heartfelt thanks to the University of Padova for funding this Erasmus research fellowship in Germany, and to Prof.ssa Alessandra Bertoldo for her availability and support as the internal supervisor.