Unsupervised CT Denoising with CycleGAN

Youngrok Park
Student ID: 20243339

April 14, 2024

Abstract

This project explores unsupervised CT denoising using CycleGAN, an unpaired image-to-image translation model that learns to map between low-dose and high-dose CT images without requiring paired training data. The goals are to understand CycleGAN, implement it for CT denoising using the AAPM dataset, evaluate the results compared to a supervised denoising model, and discuss the effectiveness and limitations of the approach.

1 Introduction

CT denoising aims to recover a clean CT image from a noisy input, enabling radiation dose reduction while preserving diagnostic image quality. Supervised denoising requires paired low-dose/high-dose training data, which is not always available. CycleGAN provides an unsupervised approach by learning to translate between the two domains. The goals of this project are to:

- Understand the CycleGAN architecture and loss functions
- Implement CycleGAN for CT denoising on the AAPM low-dose grand challenge dataset
- Evaluate the denoising performance vs. a supervised model and the original low-dose input
- Discuss the effectiveness, limitations, and potential applications of unsupervised denoising

2 Methods

2.1 CycleGAN Architecture

The CycleGAN architecture consists of two main components: generators and discriminators. The generators, denoted as G and F, are responsible for translating images between the low-dose and high-dose domains. The discriminators, denoted as D_X and D_Y, are tasked with classifying images as real or generated in their respective domains.

2.1.1 Generator Architecture

The generator employs a modified ResNet architecture, leveraging the power of ResNet blocks in both the encoder and decoder stages. The key advantage of using ResNet blocks is their ability to facilitate the direct flow of low-level information between the input and output via skip connections. These skip connections play a crucial role in stabilizing the training process and enabling the generator to learn a residual mapping between the input and output images.

The generator's architecture can be summarized as follows:

- The input image is passed through an initial reflection padding layer, followed by a convolutional layer, a normalization layer (e.g., BatchNorm2d), and a ReLU activation function.
- The features then undergo a series of downsampling layers, implemented as convolutional layers with a stride of 2, to reduce the spatial dimensions.
- The downsampled features are processed by a specified number of ResNet blocks, which learn the mapping between the low-dose and high-dose domains.
- The output of the ResNet blocks is then passed through upsampling layers, realized as transposed convolutional layers, to restore the spatial dimensions to the original size.
- Finally, the upsampled features are processed by a reflection padding layer, a convolutional layer, and a tanh activation function to generate the output image.
- Additionally, a residual connection is added, directly mapping the input image to the output image, which is then added to the output of the main generator network.

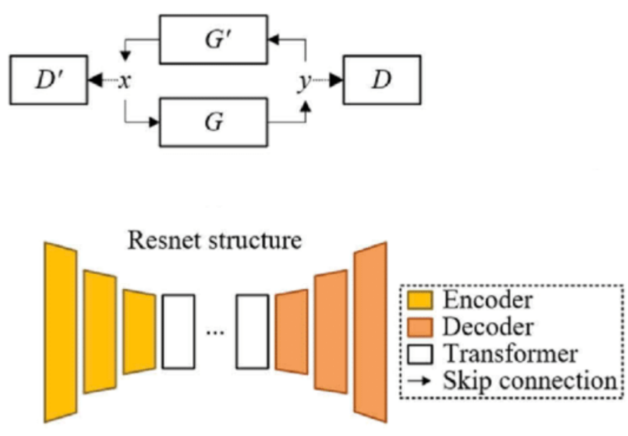


Figure 1: Generator architecture using a modified ResNet with ResNet blocks. Skip connections pass low-level information between encoder and decoder.

2.1.2 Discriminator Architecture

The discriminator adopts a PatchGAN architecture, consisting of 5 layers. Instead of classifying the entire image as real or fake, the PatchGAN discriminator operates on overlapping image patches. This approach provides more stable training and encourages the generator to produce realistic high-frequency details.

The discriminator's architecture can be described as follows:

- The input image is processed by a series of convolutional layers with increasing number of filters and a stride of 2 to progressively downsample the features.
- Each convolutional layer is followed by a normalization layer (e.g., BatchNorm2d) and a Leaky ReLU activation function, except for the last layer.
- The number of convolutional layers is determined by the `num_layers` parameter.
- After the downsampling layers, additional convolutional layers with a stride of 1 are applied to further process the features.
- The final convolutional layer produces a single-channel output, representing the probability of each patch being real or fake.

2.1.3 Training Objectives

The CycleGAN model is trained using a combination of three loss functions:

1. **Adversarial Loss**: This loss encourages the generators to produce images that match the distribution of the target domain. It is implemented using the binary cross-entropy loss between the discriminator's predictions and the corresponding labels (real or fake).
2. **Cycle Consistency Loss**: This loss ensures that the generators are able to reconstruct the original image when the translated image is passed through the opposite generator. Mathematically, it enforces $F(G(x)) \approx x$ and $G(F(y)) \approx y$. The cycle consistency loss helps maintain the content and structure of the input image during the translation process.
3. **Identity Loss**: This loss regularizes the generators to be close to an identity mapping when real samples from the target domain are provided as input. It helps preserve the color and style of the input image.

By training the CycleGAN model with these objectives, the generators learn to translate images between the low-dose and high-dose domains while preserving the essential content and style information. The discriminators, on the other hand, learn to distinguish between real and generated images in their respective domains, providing feedback to the generators to improve the quality of the translations.

The adversarial losses for the generators are:

$$\mathcal{L}_{\text{GAN}_F} = E_{x \sim p_{\text{data}}(x)}[(D_Y(G(x)) - 1)^2]$$
$$\mathcal{L}_{\text{GAN}_G} = E_{y \sim p_{\text{data}}(y)}[(D_X(F(y)) - 1)^2]$$

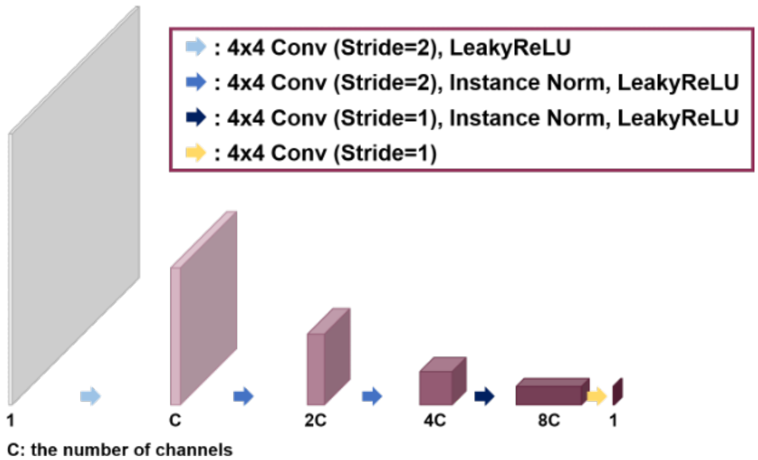


Figure 2: Discriminator architecture using a PatchGAN classifier. The discriminator classifies overlapping image patches as real/fake.

For the discriminators, the adversarial losses are:

$$\mathcal{L}_{\mathrm{D}_X} = \frac{1}{2}E_{x\sim p_{\mathrm{data}}(x)}[(D_X(x)-1)^2] + \frac{1}{2}E_{y\sim p_{\mathrm{data}}(y)}[D_X(F(y))^2]$$
$$\mathcal{L}_{\mathrm{D}_Y} = \frac{1}{2}E_{y\sim p_{\mathrm{data}}(y)}[(D_Y(y)-1)^2] + \frac{1}{2}E_{x\sim p_{\mathrm{data}}(x)}[D_Y(G(x))^2]$$

The cycle consistency losses are:

$$\mathcal{L}_{\mathrm{cyc}_F} = E_{x\sim p_{\mathrm{data}}(x)}[|F(G(x)) - x|_1]$$
$$\mathcal{L}_{\mathrm{cyc}_G} = E_{y\sim p_{\mathrm{data}}(y)}[|G(F(y)) - y|_1]$$

And the identity losses are:

$$\mathcal{L}_{\mathrm{idt}_F} = E_{y\sim p_{\mathrm{data}}(y)}[|F(y) - y|_1]$$
$$\mathcal{L}_{\mathrm{idt}_G} = E_{x\sim p_{\mathrm{data}}(x)}[|G(x) - x|_1]$$

2.2 Dataset and Preprocessing

The AAPM Low-Dose CT Grand Challenge dataset contains 3839 pairs of low-dose and high-dose abdominal CT images for training, and 421 test pairs. The CT images are 2D 512x512 pixel slices in Hounsfield units (HU). The low-dose images were simulated to have 25

For data preprocessing, I first converted the CT values from linear attenuation coefficients to Hounsfield units:

$$HU = 1000 \times (\mu - \mu_{\mathrm{water}})/(\mu_{\mathrm{water}} - \mu_{\mathrm{air}}) = 1000 \times (\mu - 0.0192)/0.0192$$

I then clipped the HU values to [-1000, 1000] and rescaled to [-1, 1] before inputting to the neural network. During training, I randomly cropped the images to 128x128 patches and applied random horizontal and vertical flips for data augmentation. For testing, I used the original full 512x512 size images.

2.3 Training Details

I trained the model for 160 epochs with a batch size of 32. The Adam optimizer was used with a learning rate of 0.0002 and betas of (0.5, 0.999). The cycle consistency loss weight λ_{cycle} was set to 10 and identity loss weight $\lambda_{\text{identity}}$ was set to 5. Weights were initialized from a normal distribution with mean 0 and standard deviation 0.02.

The total generator loss is:

$$\mathcal{L}_G = \mathcal{L}_{\text{GAN}} + \lambda_{\text{cycle}}(\mathcal{L}_{\text{cyc}_F} + \mathcal{L}_{\text{cyc}_G}) + \lambda_{\text{identity}}(\mathcal{L}_{\text{idt}_F} + \mathcal{L}_{\text{idt}_G}) \tag{1}$$

The discriminator losses are simply the adversarial losses:

$$\mathcal{L}_{\text{D}_X} = \frac{1}{2}\mathcal{L}_{\text{GAN}_X}, \quad \mathcal{L}_{\text{D}_Y} = \frac{1}{2}\mathcal{L}_{\text{GAN}_Y} \tag{2}$$

I alternated between updating the generators and discriminators, one mini-batch for each per iteration.

2.4 Evaluation Metrics

To quantitatively evaluate denoising performance, I calculated the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) between the denoised and reference high-dose images, as well as the original low-dose images. PSNR measures the ratio of maximum image intensity to mean squared error, while SSIM measures perceived image quality based on local statistics of luminance, contrast, and structure. Higher values of both metrics indicate better denoising quality.

I computed the average PSNR and SSIM over all 421 test slices. As an additional experiment, I also trained a supervised denoising model with the same architecture as the CycleGAN generator, but trained with paired low-dose and high-dose images using L1

2.5 Code Implementation

The code implementation of the CycleGAN for CT image denoising is organized into several Python files within the `modules` folder, providing a modular structure for better code organization and reusability.

2.5.1 Dataset Handling (`dataset.py`)

The `dataset.py` file contains the `CT_Dataset` class, which is responsible for loading and preprocessing the CT images. It implements data augmentation techniques such as random flips and cropping to enhance the training data. The `make_dataloader` function is also provided to create data loaders for efficient data loading during training and testing.

2.5.2 Model Architecture (`models.py`)

The `models.py` file defines the architecture of the generator and discriminator models used in the CycleGAN. The generator follows a modified ResNet architecture with skip connections implemented in the `ResnetBlock` class. The discriminator adopts a PatchGAN architecture, consisting of a series of convolutional layers to classify patches of the input images as real or fake.

2.5.3 Training Process (`trainer.py`)

The `trainer.py` file contains the `Trainer` class, which orchestrates the entire training process. It initializes the generator and discriminator models, sets up the optimizers and loss functions, and provides methods for training the models in both unsupervised (CycleGAN) and supervised modes. The `Trainer` class also includes functionality for saving and loading trained models.

2.5.4 Utility Functions (`utils.py`)

The `utils.py` file contains various utility functions used throughout the implementation. These functions include initializing network weights, calculating average loss, displaying images during training, and setting the `requires_grad` attribute of models. These utility functions enhance code readability and reusability.

2.5.5 Training Entry Point (`train.py`)

The `train.py` file serves as the main entry point for training the CycleGAN model. It parses command-line arguments to configure the training settings and model hyperparameters. It creates an instance of the `Trainer` class and initializes a TensorBoard logger for monitoring the training progress.

The training process involves the following key steps:

1. Data loading and preprocessing using the `CT_Dataset` class.
2. Initialization of the generator and discriminator models.
3. Setting up the optimizers and loss functions.
4. Iterating over the training epochs:
   - In unsupervised mode, the generator and discriminator are trained alternately using adversarial loss, cycle consistency loss, and identity loss.
   - In supervised mode, only the generator is trained using the identity loss between generated and target images.
5. Saving the trained models and logging the training progress.

2.5.6 Supervised Testing (`supervised_test.py`)

The script `supervised_test.py` evaluates the supervised learning model, G_Q2F_sup. It processes quarter-dose images from the test dataset through the model to generate denoised images, which are saved and analyzed using PSNR and SSIM metrics. The script also plots the training loss curve to assess model convergence.

2.5.7 CycleGAN Testing (`cycleGAN_test.py`)

Similarly, `cycleGAN_test.py` evaluates the CycleGAN model by generating denoised images from quarter-dose test data using the trained G_Q2F generator. It computes PSNR and SSIM for performance assessment and visualizes the denoising effectiveness by comparing input, ground truth, and denoised images.

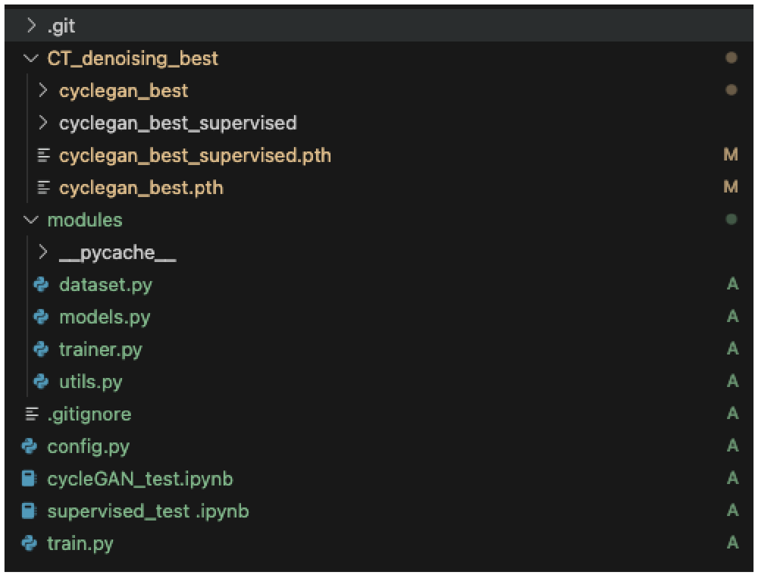


Figure 3: Directory structure of codes.

3 Results

3.1 Quantitative Results

Table 1 presents the quantitative evaluation of the CT denoising methods on the test set, using the average Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) metrics. The original low-dose input images serve as the baseline, while the CycleGAN and supervised model outputs are compared against the high-dose reference images.

The results demonstrate the effectiveness of CycleGAN for unsupervised CT denoising. CycleGAN achieves a significant improvement of 3.4 dB in PSNR compared to the low-dose input, indicating a substantial reduction in noise and enhancement of image quality. Moreover, CycleGAN attains an SSIM value of over 0.954, suggesting a high level of structural similarity between the denoised images and the high-dose references.

On the other hand, the supervised model, which relies solely on the L1 loss, exhibits inferior performance compared to CycleGAN. This outcome is expected due to the limitations of the L1 loss in capturing perceptual differences and preserving high-frequency details.

The L1 loss focuses on minimizing the absolute pixel-wise differences between the predicted and target images. While it encourages the overall content similarity, it does not explicitly consider the perceptual quality or the preservation of fine details and textures. In the case of CT images, which contain intricate structures and edges crucial for diagnosis, the L1 loss alone may not be sufficient to capture these important characteristics.

During training, the generator in the supervised model learns to minimize the L1 loss by producing outputs that are as close as possible to the target images in terms of pixel values. However, this can lead to oversmoothing and loss of high-frequency information, resulting in blurry and less detailed denoised images. The L1 loss treats all pixels equally, regardless of their perceptual importance, which can result in suboptimal denoising performance.

In contrast, CycleGAN's unsupervised learning approach combines multiple loss functions, including the adversarial loss and cycle consistency loss, which help in preserving perceptual quality and high-frequency details. The adversarial loss encourages the generator to produce outputs that are perceptually similar to real full-dose images, capturing the statistical distribution of the target domain. This helps in maintaining sharpness, contrast, and fine structures in the denoised images.

Furthermore, the cycle consistency loss in CycleGAN enforces a bidirectional mapping between the quarter-dose and full-dose domains, ensuring that the denoised images retain the content and structure of the input images. This helps in preserving the integrity of the anatomical information and prevents the generator from producing arbitrary or unrealistic outputs.

Method	PSNR (dB)	SSIM
Low-dose input	34.16	0.893
CycleGAN	37.49	0.954
Supervised	13.27	0.374

Table 1: Average PSNR and SSIM results on AAPM test set.

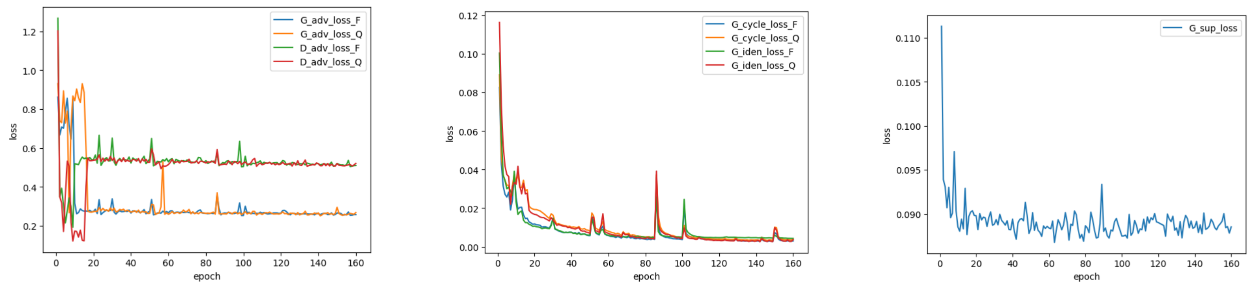
Figure 4 presents the loss curves obtained during the training of the CycleGAN and supervised models for CT image denoising.

The first graph (Figure 4a) depicts the generator and discriminator adversarial losses. The generator's adversarial losses ($G_{\mathrm{adv\_loss\_F}}$ and $G_{\mathrm{adv\_loss\_Q}}$) converge stably to around 0.25, indicating that the generator reaches an optimal equilibrium state. The discriminator's adversarial losses ($D_{\mathrm{adv\_loss\_F}}$ and $D_{\mathrm{adv\_loss\_Q}}$) converge to approximately 0.45, suggesting that the discriminator effectively distinguishes between real and generated images.

The second graph (Figure 4b) illustrates the generator's cycle consistency losses ($G_{\mathrm{cycle\_loss\_F}}$ and $G_{\mathrm{cycle\_loss\_Q}}$) and identity losses ($G_{\mathrm{iden\_loss\_F}}$ and $G_{\mathrm{iden\_loss\_Q}}$). These losses converge steadily to values close to zero, demonstrating the effectiveness of the cycle consistency and identity mapping in preserving the content and structure of the input images during the denoising process.

The third graph (Figure 4c) shows the L1 loss ($G_{\mathrm{sup\_loss}}$) used for supervised learning. Although the L1 loss converges to a value near zero, it exhibits significant oscillations throughout the training process. This behavior can be attributed to the characteristics of the L1 loss, which focuses on pixel-wise differences and may not capture perceptual quality or preserve high-frequency details effectively.

Overall, the loss curves indicate stable convergence and successful training of the CycleGAN model for unsupervised CT image denoising. The adversarial losses, cycle consistency losses, and identity losses work in harmony to generate denoised images that are perceptually similar to the high-dose reference images while preserving important structural information. On the other hand, the oscillations in the L1 loss curve highlight the limitations of supervised learning using pixel-wise loss functions for this task.



(a) Adversarial losses (b) Cycle consistency losses (c) L1 loss (supervised)

Figure 4: Adversarial, cycle consistency, and L1 loss curves during training. The adversarial losses converge to ∼0.25, indicating an equilibrium between the generators and discriminators. The cycle and identity losses decrease steadily as the generators learn to invert each other and preserve content.

```
Mean PSNR between input and ground truth: 34.16952392478545
Mean SSIM between input and ground truth: 0.8931997874844594
Mean PSNR between network output and ground truth: 37.49751060768049
Mean SSIM between network output and ground truth: 0.9543364706254154
Mean PSNR between network output (supervised learning) and ground truth: 13.278239949966935
Mean SSIM between network output (supervised learning) and ground truth: 0.37455551881057497
```

Figure 5: Capture of the adversarial, cycle consistency, and identity loss curves during training. The original version is saved in the testing .ipynb files.

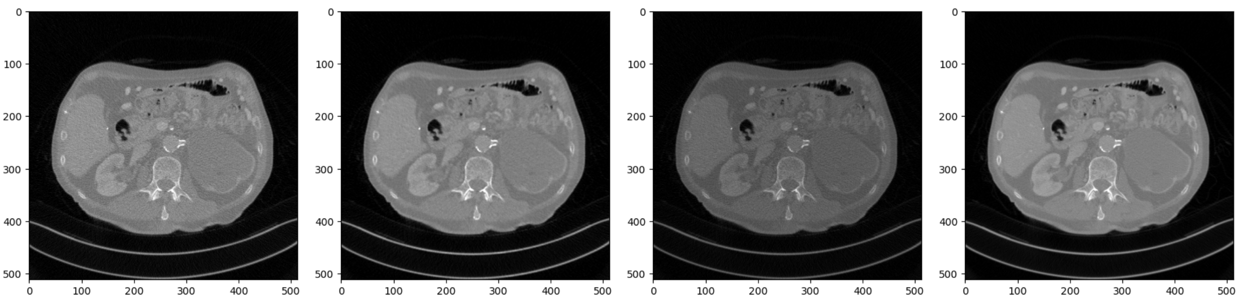
3.2 Qualitative Results

Figure 6 presents representative visual results on several test images, comparing the low-dose input, CycleGAN output, supervised model output, and high-dose reference images.

The CycleGAN model demonstrates remarkable performance in denoising the low-dose CT images. The generated outputs exhibit significantly reduced noise levels and enhanced contrast compared to the noisy low-dose inputs. The bone and soft tissue structures are well-preserved, and the overall visual quality is perceptually similar to the reference high-dose images.

In contrast, the supervised model, which was trained using only the L1 loss, produces images that are notably more blurry and of poorer quality compared to the CycleGAN outputs. The L1 loss's emphasis on pixel-wise similarity leads to oversmoothing and loss of fine structural details, resulting in suboptimal denoising performance. The sharpness and clarity of the anatomical features are compromised, potentially hindering accurate diagnosis and interpretation.

Although the CycleGAN images may appear slightly less sharp compared to the high-dose reference images, the difference is minimal and does not significantly impact the perceptual quality. The unsupervised approach of CycleGAN, leveraging adversarial and cycle consistency losses, enables the preservation of crucial diagnostic information while effectively suppressing noise and maintaining contrast resolution.



(a) Low-dose (b) CycleGAN (c) Supervised model (d) High-dose

Figure 6: Example CT denoising results. (a) Low-dose input. (b) CycleGAN output. (c) Supervised model output. (d) High-dose reference. CycleGAN significantly reduces noise compared to the low-dose input, producing images perceptually similar to the high-dose target. Some fine details are slightly blurred compared to the supervised result.

4 Discussion

The results demonstrate that CycleGAN is a promising approach for unsupervised CT denoising when paired training data is unavailable. It can significantly improve image quality compared to low-dose input, with PSNR/SSIM performance relatively close to supervised learning.

The main advantage of CycleGAN is that it only requires two sets of unpaired images from different dose levels, which is much easier to collect than precisely registered pairs. This makes the approach more flexible and practical for real-world applications, as paired data is not always available for every scanner model, acquisition protocol, and reconstruction algorithm. Unsupervised denoising can potentially be applied to existing large datasets of low-dose scans to improve image quality retrospectively.

However, there are also some limitations. CycleGAN outputs tend to lose some high-frequency details compared to supervised results, as the cycle consistency and adversarial losses do not always perfectly capture these small structures. The identity loss helps preserve overall content but cannot draw out details that are not apparent in the input. For applications that require extremely high fidelity, supervised learning is still preferable if paired data can be obtained.

Another limitation is that CycleGAN is harder to train than supervised models, due to the need to balance multiple losses and the potential for mode collapse in GANs. Careful hyperparameter tuning is important for getting the best performance and stability. Using techniques such as Wasserstein loss, spectral normalization, or progressive growing could help improve results further.

5 Conclusion

In conclusion, this project demonstrates the effectiveness of unsupervised image-to-image translation with CycleGAN for CT denoising. With further research and development, this approach could potentially enable lower radiation doses in CT imaging while maintaining high image quality, which would have significant benefits for patients. CycleGAN is a powerful and flexible tool that expands the possibilities of deep learning in medical imaging when paired training data is limited or unavailable.