

國立臺灣大學醫學院醫療器材與醫學影像研究所



碩士論文

Institute of Medical Device and Imaging

College of Medicine

National Taiwan University

Master's Thesis

利用卷積神經網路恢復 X 光電腦斷層掃描影像之空間
解析度

Spatial Resolution Restoration of X-ray Computed
Tomography Images Using Convolutional Neural Network

林俊丞

Jun-Cheng Lin

指導教授：吳文超 博士

Advisor: Wen-Chau Wu, Ph.D.

中華民國 114 年 6 月

June, 2025



國立臺灣大學碩士學位論文

口試委員會審定書

MASTER'S THESIS ACCEPTANCE CERTIFICATE
NATIONAL TAIWAN UNIVERSITY

論文中文題目：利用卷積神經網路恢復 X 光電腦斷層掃描影像之空間解析度

論文英文題目：Spatial Resolution Restoration of X-ray Computed Tomography
Images Using Convolutional Neural Network

本論文係 林俊丞 (R12458008) 在國立臺灣大學醫療器材與醫學影像研究所完成之碩士學位論文，於民國 114 年 6 月 30 日承下列考試委員審查通過及口試及格，特此證明。

The undersigned, appointed by the Department / Graduate Institute of Institute of Medical Device and Imaging on 30 June, 2025 have examined a Master's Thesis entitled above presented by Jun-Cheng Lin (R12458008) candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

<u>張子強</u>	_____	<u>劉高印</u>
(指導教授 Advisor)		
<u>張子強</u>	<u>林益如</u>	<u>李國輝</u>

系（所、學位學程）主管 Director: _____





Acknowledgements

回首這段研究與寫論文的日子，過程中有挑戰、有迷惘，也有成就與感動。能夠走到這一步，我衷心感謝一路上遇見的每一位貴人。

首先，要特別感謝我的指導教授吳文超教授。從題目的發想、資料的準備，甚至到寫作，吳老師都給了我非常多的指導與實質協助，不只是學術上的引領，也讓我在這段研究旅程中學會了許多事情。接著也感謝四位口試委員鍾孝文教授、林益如教授、蔡炳輝教授以及劉高郎醫師，謝謝您們撥冗審閱我的論文，並提供寶貴的建議與指教，讓本研究更加完整、深刻。此外，也特別感謝張晉誠醫師在研究期間給予我許多專業上的協助，無論是在資料理解還是臨床意義上，都是推動這篇論文能夠實際落地的重要力量。

還有也要感謝實驗室的成員：士洋、帛誼、鈞唯，謝謝你們在每次 meeting 時總是可以給我許多反饋以及建議，讓我受益良多。

最後，還要感謝我的家人們，謝謝你們一直在背後默默支持我。



摘要

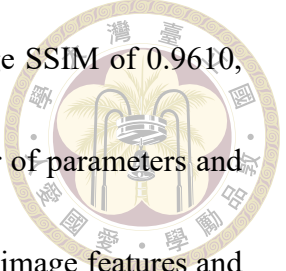
薄切片電腦斷層掃描 (CT) 切片影像能提供比傳統 CT 切片影像更豐富的診斷細節，但是臨床上為了配合儲存、傳輸及重建等流程規格，通常只能取得較厚的切片影像，限制了醫師對細微結構的觀察。為解決此挑戰，本研究開發一種深度學習方法，從現有的厚切 CT 影像恢復出高品質的薄切影像。這個方法是基於卷積神經網路 (CNN) 的框架，其特色在於融合軸向與冠狀兩個視角的厚切影像資訊作為輸入，並採用殘差學習架構，透過數據訓練學習從厚切面到薄切面的非線性映射。研究中設計了多個不同參數數量的 2D CNN 模型，以傳統插值方法為基準線，並使用結構相似性指數 (SSIM) 及視覺評估進行性能衡量。實驗結果系統性地驗證了本研究所提出的雙視角 2D CNN 模型能有效重建薄切影像，其中，高參數量 2D CNN 模型在 SSIM 指標上表現最佳，平均 SSIM 值為 0.9610，顯著優於使用與此 2D 架構相同變數量的 3D 模型以及傳統插值方法，更能捕捉影像特徵與保持空間連續性。我們透過像素級與視覺評估呈現模型恢復邊界、紋理及細節的能力，更進一步以模擬實驗探索模型所能夠恢復的切片厚度範圍，作為影像資料壓縮效率的參考。本研究成功驗證了利用雙視角深度學習重建高品質薄切片 CT 影像的可行性與有效性，為前述臨床問題提供了一個解決方案。

關鍵字：薄切片、電腦斷層掃描、影像恢復、深度學習、雙視角



Abstract

Thin-slice computed tomography (CT) images provide richer diagnostic details as compared with conventional CT images, yet clinical practice often relies on thicker slices due to the limits in storage capacity, transmission bandwidth, and reconstruction processes, hampering radiologists from observing fine structures. To address this challenge, this study developed and evaluated a deep learning method to restore high-quality thin-slice images from existing thick-slice CT data. We proposed to use convolution neural networks (CNNs) to fuse thick-slice image information from both axial and coronal views. The model employed a residual learning architecture to extract the non-linear mapping from thick to thin slices. A series of 2D CNN models with varied numbers of parameters were designed for comparison, using traditional interpolation methods as a baseline, with performance measured by the structural similarity index (SSIM) and visual evaluation. Experimental results systematically validated the effectiveness of the proposed dual-view 2D CNN models in restoring thin-slice images. Of the 2D CNN models constructed, the



high-parameter model achieved the best performance with an average SSIM of 0.9610, significantly outperforming the 3D model with a comparable number of parameters and traditional interpolation methods. The optimal model better captured image features and preserved spatial continuity. Pixel-level and visual evaluations also demonstrated the model' s ability to restore boundaries and textures. Further, we performed numerical simulation to evaluate the slice thickness recoverable with our model (i.e., compression efficiency). In conclusion, this study has successfully validated the feasibility and effectiveness of using dual-view deep learning to restore high-quality thin-slice CT images, which offers a potential solution to the abovementioned clinical problem.

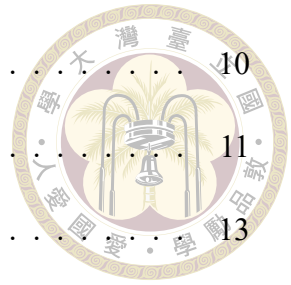
Keywords: Thin-slice, Computed Tomography, Image Restoration, Deep Learning, Dual-View



Contents

	Page
Verification Letter from the Oral Examination Committee	i
Acknowledgements	ii
摘要	iii
Abstract	iv
Contents	v
List of Figures	viii
List of Tables	x
Chapter 1 Introduction	1
1.1 Background	1
1.2 Motivation and Problem Statement	1
1.3 Existing Methods and Limitations	2
1.4 Study Purpose	4
Chapter 2 Materials and Methods	5
2.1 Image Data	7
2.2 Image Preprocessing	7
2.2.1 Normalization	8
2.2.2 Resizing	8

2.2.3	Cropping	10
2.3	Deep Learning Model Structure	11
2.3.1	Feature Extraction	13
2.3.2	Feature Fusion	15
2.3.3	Residual Learning	15
2.3.4	Loss Function	17
2.3.5	Training Settings	19
Chapter 3	Results and Discussion	22
3.1	Training Results	22
3.2	Model Performance	27
3.3	Model Comparison	29
3.4	Performance Comparison of Single-View and Dual-View Models . .	37
3.5	Simulation	40
3.6	Study Limitations	44
Chapter 4	Conclusion	45
	References	45

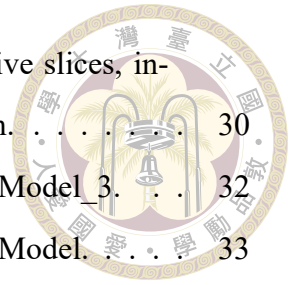




List of Figures

2.1	The overall flowchart of the proposed method, including preprocessing, model training, and output conversion.	6
2.2	Illustration of the five-fold slice interpolation process to achieve a $1 \times 1 \times 1 \text{ mm}^3$ effective resolution. (a) depicts the process for original axial images, and (b) for axial images reconstructed from original coronal images.	9
2.3	Illustration of potential size and alignment mismatches when reconstructing axial views from coronal data before specific alignment processing. .	10
2.4	Illustration of preprocessing results ready for model input.	11
2.5	The overall architecture of the proposed deep learning model, highlighting the feature extraction, fusion, and residual learning stages.	12
2.6	The detailed structure of the Feature Extraction Stage. Axial and coronal inputs are processed through separate convolutional pathways before feature map concatenation.	14
2.7	The detailed structure of a single residual block used in the Residual Learning Stage.	16
3.1	Training and validation loss curves for 2D_Model_1.	23
3.2	Training and validation loss curves for 2D_Model_2.	24
3.3	Training and validation loss curves for 2D_Model_3.	24
3.4	Training and validation loss curves for 2D_Model_4.	24
3.5	Training and validation loss curves for 2D_Model_5.	25
3.6	Training and validation loss curves for 2D_Model_6.	25
3.7	Training and validation loss curves for 3D_Model.	25

3.8	Restoration results of 2D_Model_1 showing five consecutive slices, including axial/coronal inputs, model output, and ground truth.	30
3.9	Comparison of restored images across 2D_Model_1 to 2D_Model_3.	32
3.10	Comparison of restored images across 2D_Model_4 to 3D_Model.	33
3.11	Pixel intensity profile using 2D_Model_1.	35
3.12	Example of pixel intensity profile showing artifacts in the restored image generated by 2D_Model_1.	36
3.13	Pixel intensity profile along a selected row in the coronal plane using 2D_Model_1.	37
3.14	Visual comparison of thin-slice CT image restoration. The dual-view approach demonstrates superior detail recovery and structural integrity.	39
3.15	Illustration of the simulation pipeline: 1mm ground truth (GT) images are averaged to create 10mm, 15mm, or 20mm thick simulated inputs. These inputs are then processed by 2D_Model_1 to produce a restored 1mm resolution output.	41
3.16	Visual comparison of 2D_Model_1 restoration performance on simulated thick-slice inputs.	43





List of Tables

2.1	Hyperparameter settings for the Adam optimizer.	20
2.2	Training Hyperparameter details.	20
2.3	Training Environment details.	21
3.1	Residual Block channel sequence and parameter count for all models. The sequence shows the number of output channels after each of the 9 residual blocks, starting from the initial input channels to Block 1.	23
3.2	Training time and GPU memory cost for each model.	23
3.3	SSIM values of each model across different test data.	27
3.4	SSIM values of traditional interpolation methods across test datasets. . . .	28
3.5	Comparison of SSIM values for single-view models and the 2D_Model_1 across test datasets.	38
3.6	SSIM performance of simulated inputs restored by 2D_Model_1	42



Chapter 1 Introduction

1.1 Background

Thin-slice x-ray computed tomography (CT) [1] enhances diagnostic quality by providing better spatial resolution as compared with conventional CT images where slice thickness is usually larger than in-plane pixel sizes. Thin slices mitigate partial volume effect, allow for clearer visualization of lesion boundaries and fine anatomical structure, and thus improve overall diagnostic accuracy and subsequent treatment planning. Thin slices also allow for isotropic spatial resolution which improves the quality of volume rendering when necessary.

1.2 Motivation and Problem Statement

Although thin-slice images provide rich details, they present practical challenges in clinical workflows. The raw data volume of thin-slice images is extremely large, demanding significant storage space in Picture Archiving and Communication Systems (PACS) and high bandwidth for network transmission. Furthermore, modern CT scanners typically perform automated post-processing after the scan, and reconstruct the raw thin-slice data into standardized thick slices (e.g., 5 mm) according to predefined clinical proto-

cols. This reconstruction step is often integral to the standard image generation process for radiologists' interpretation.



Consequently, due to these storage, transmission, and standardized reconstruction procedures, the images archived long-term in PACS and routinely accessed by clinicians are often the processed, standard-thickness images. The original thin-slice data, containing the most complete details, might be discarded shortly after reconstruction or not stored routinely. This creates a clinical limitation: radiologists/clinicians often cannot retrieve the original thin-slice images to examine minute lesions or anatomical structures.

Under these restrictions, restoration of thin-slice quality images from existing thick-slice CT images – without calling-back patients for extra scanning or searching for potentially unavailable raw data – has emerged as a problem of great clinical and research interest.

1.3 Existing Methods and Limitations

A common method to resolution restoration is using interpolation approaches, such as nearest neighbor, bilinear, or cubic interpolation, to compute the to-be-restored pixel values by referring to neighboring voxels [2, 3]. These methods are popular because of their simplicity and practicable computational efficiency. However, directly through interpolation it is difficult to maintain structural and textural relationships among the pixels of the image, which in many situations makes them a lossy frame reconstruction and leads to image blurring, particularly in areas of significant structural variations or fine textures/features.

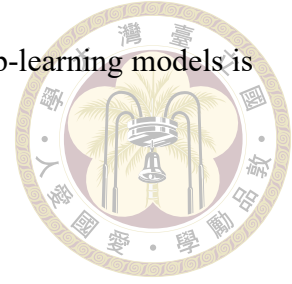
With recent breakthroughs in model architecture and computing power, deep learning

has been increasingly applied to medical image reconstruction. The advantage of convolutional neural networks (CNNs) [4] is that they can learn the nonlinear mapping between input and target images from large-scale training data, by automatically extracting multi-level features of input images from spatial, textural, and semantic information [5, 6]. This overcomes the limitations of conventional techniques which often struggle to recover high-frequency details and structural authenticity.

Recent studies implemented super-resolution [7, 8] for a single two-dimensional (2D) image by using SRCNN (Super-Resolution Convolutional Neural Network) [9] and VDSR (Very Deep Super-Resolution) [10]. These pioneering methods demonstrated notable improvements over traditional bi-cubic interpolation on benchmark natural image datasets. For instance, SRCNN achieved an average SSIM improvement of approximately 0.0472 and a PSNR gain of around 1.88 dB, while VDSR provided an average SSIM improvement of about 0.0626 and a PSNR gain of approximately 2.58 dB. Although these studies showed satisfactory resolution and visual quality in the reconstructed images, primarily for general 2D content, it is unclear whether their methods are directly applicable to three-dimensional (3D) clinical images considering that their methods did not incorporate volumetric spatial continuity and contextual dependencies crucial for 3D medical data. A later study introduced 3D CNNs to include volumetric spatial information for CT and magnetic resonance image reconstruction [11, 12]. On the other hand, multi-view input images have been used in several artificial intelligence tasks of medical images, such as segmentation and detection [13, 14], to enable the model to sense multidirectional information contained in edges and textures.

Although several deep-learning models have been proposed for CT image reconstruction [15], few considered both multi-view fusion [16] and light computation [17, 18]. It

also remained unclear how much or whether the performance of deep-learning models is better than traditional interpolation methods.



1.4 Study Purpose

To address the abovementioned gaps, we proposed a CNN-based lightweight framework to restore thin-slice CT images from both axial and coronal views of thick-slice CT images.



Chapter 2 Materials and Methods

The study flowchart is shown in Fig. 2.1 and described separately in the following subsections.

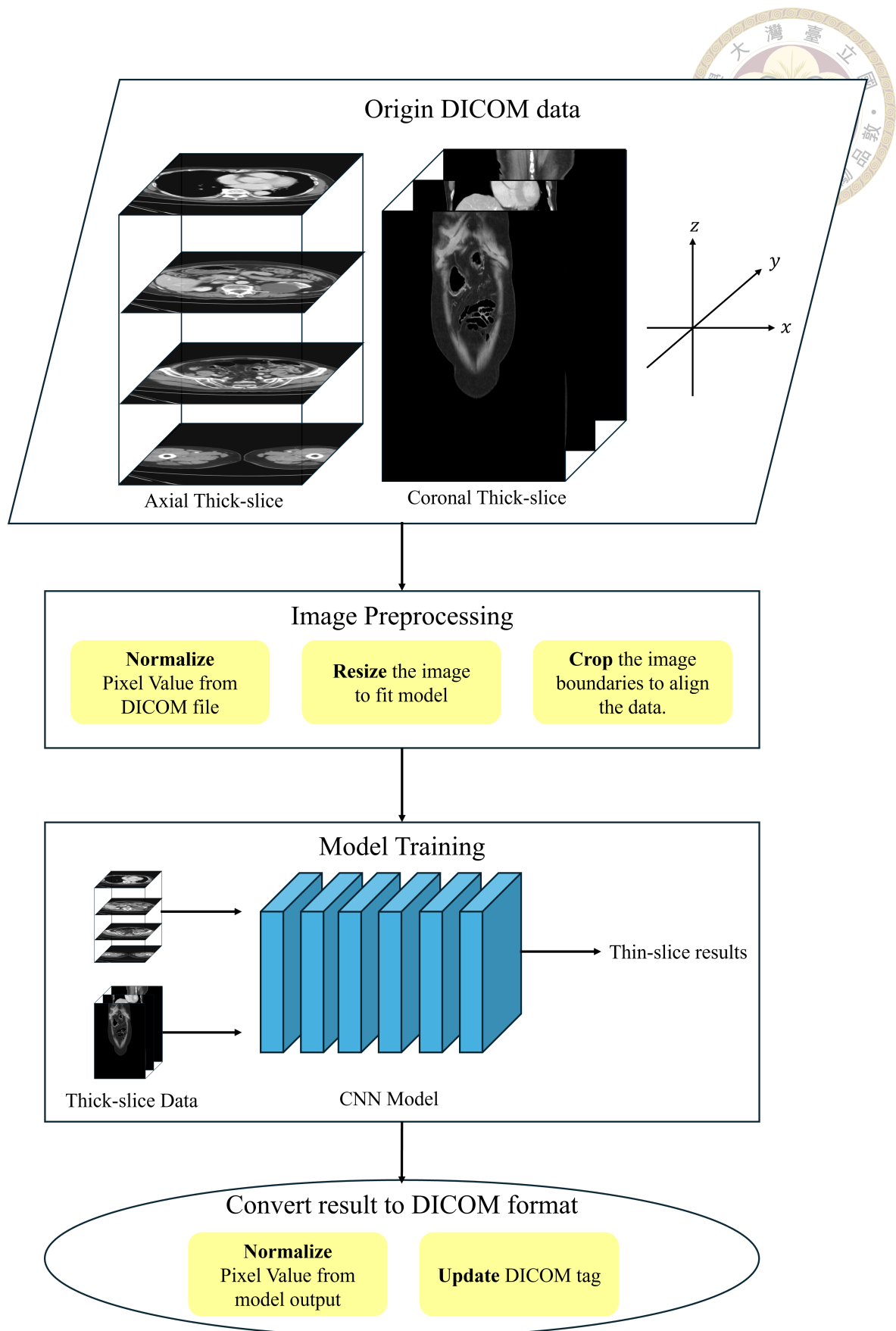


Figure 2.1: The overall flowchart of the proposed method, including preprocessing, model training, and output conversion.

2.1 Image Data



This study was conducted using the abdominal CT images acquired in National Taiwan University Hospital. A total of 109 adult subjects were initially collected in this study. The informed consent form was waived by the Research Ethics Committee for de-identified retrospective data analysis. Among them, 100 subjects with complete and high-quality scans were included for model development and evaluation, with data in the DICOM format. For each subject, the data contained thin-slice images (1 mm and isotropic) and thick-slice images (1×1 mm in-plane and 5 mm through-plane).

To build the deep-learning model, the data was divided into training, validation, and test sets (N = 90, 10, and 9). Training was performed on the training set, while the validation set was used to track model performance, optimize hyper-parameters, and prevent overfitting. The test set was reserved for final evaluation of the model's performance. The inputs were thick-slice images (including both axial and coronal views), and thin-slice images were the ground truth.

2.2 Image Preprocessing

The deep learning model takes two sets of axial slices as the input: one from the original thick-slice axial images and the other from the original thick-slice coronal images reconstructed into axial slices. The preprocessing steps are described as follows.

2.2.1 Normalization

To mitigate inconsistency in signal intensities from scan to scan, the pixel values of the DICOM images were converted to Hounsfield Units (HU):

$$HU = (\text{PixelValue} \times \text{RescaleSlope}) + \text{RescaleIntercept} \quad (2.1)$$

2.2.2 Resizing

This step was to match the slice number between the input images and the ground truth where the resolution was $1 \times 1 \times 1 \text{ mm}^3$. By referring to the coordinate system in Fig. 2.2, the original axial images (5mm thick along the z axis) underwent five-fold interpolation along the z axis. The axial images reconstructed from the original coronal images (5mm thick along the y axis) underwent five-fold interpolation along the y axis.



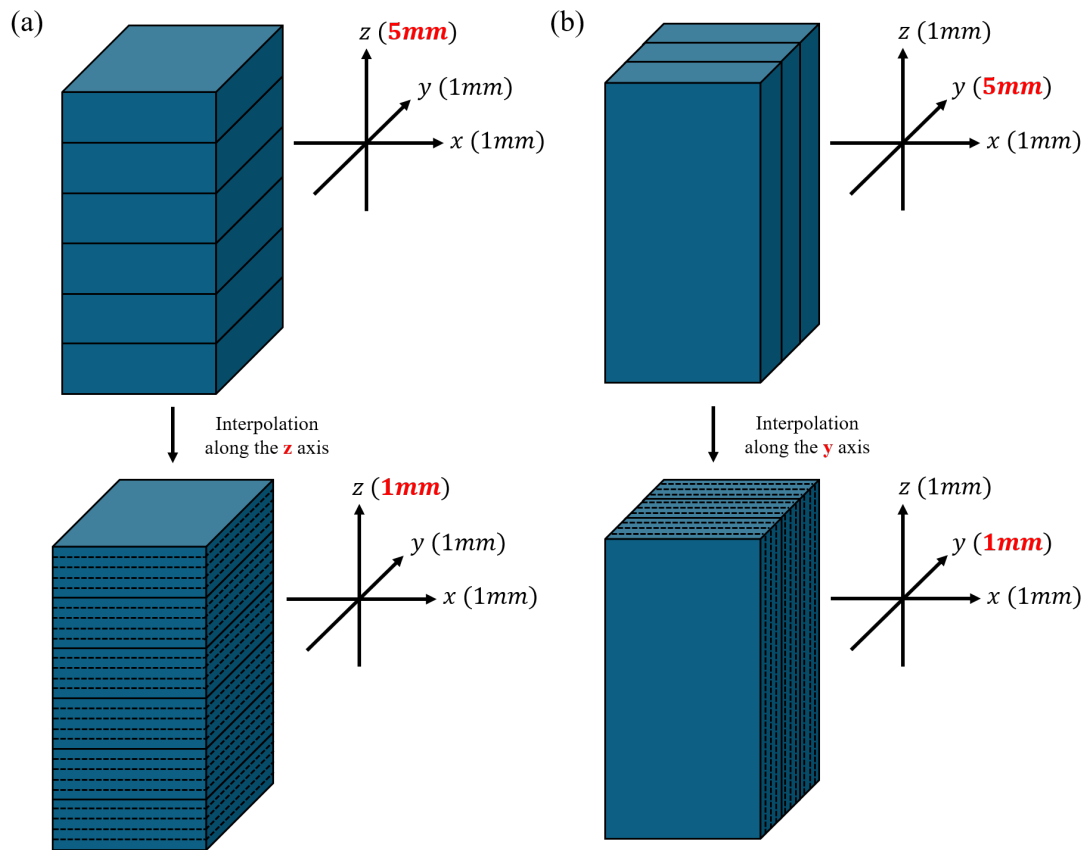


Figure 2.2: Illustration of the five-fold slice interpolation process to achieve a $1 \times 1 \times 1 \text{ mm}^3$ effective resolution. (a) depicts the process for original axial images, and (b) for axial images reconstructed from original coronal images.

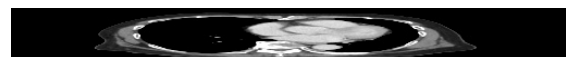


2.2.3 Cropping

Coronal images are first reconstructed into a series of axial slices. This direct reconstruction can lead to size and alignment mismatches with the original axial images, as illustrated in Fig. 2.3. To resolve this, a process of detection, cropping, resizing, and padding is applied, primarily targeting the left and right edges of the coronally-reconstructed axial images, as revealed through image analysis. This procedure ensures that the resulting axial images derived from coronal scans are of the same size and spatial position as the original axial images from the thick-slice acquisition. Such processing guarantees high similarity between the coronally-reconstructed axial images and the original axial planes, boosting the stability and accuracy of model training. The final appearance of the preprocessed input images is illustrated in Fig. 2.4.



(a) Original thick-slice axial image.



(b) Axial image reconstructed from coronal view, showing potential mismatch.

Figure 2.3: Illustration of potential size and alignment mismatches when reconstructing axial views from coronal data before specific alignment processing.



Figure 2.4: Illustration of preprocessing results ready for model input.

2.3 Deep Learning Model Structure

After image preprocessing, the resulting CT images serve as inputs for the image restoration model developed in this study. Our work primarily focuses on a 2D Convolutional Neural Network (2D-CNN) architecture that incorporates Residual Learning to enhance the restoration of fine image details. A key aspect of our model is its simultaneous use of CT images derived from both the axial and coronal thick-slice scans as input, enabling the network to learn and fuse information from multiple directions. Through multiple convolution operations and strategic feature fusion, the model is designed to learn intricate textures and structural details within the images.

The overall structure of the model, illustrated in Fig. 2.5, can be conceptually divided into three main stages: a Feature Extraction Stage, a Feature Fusion Stage, and a Residual Learning Stage.

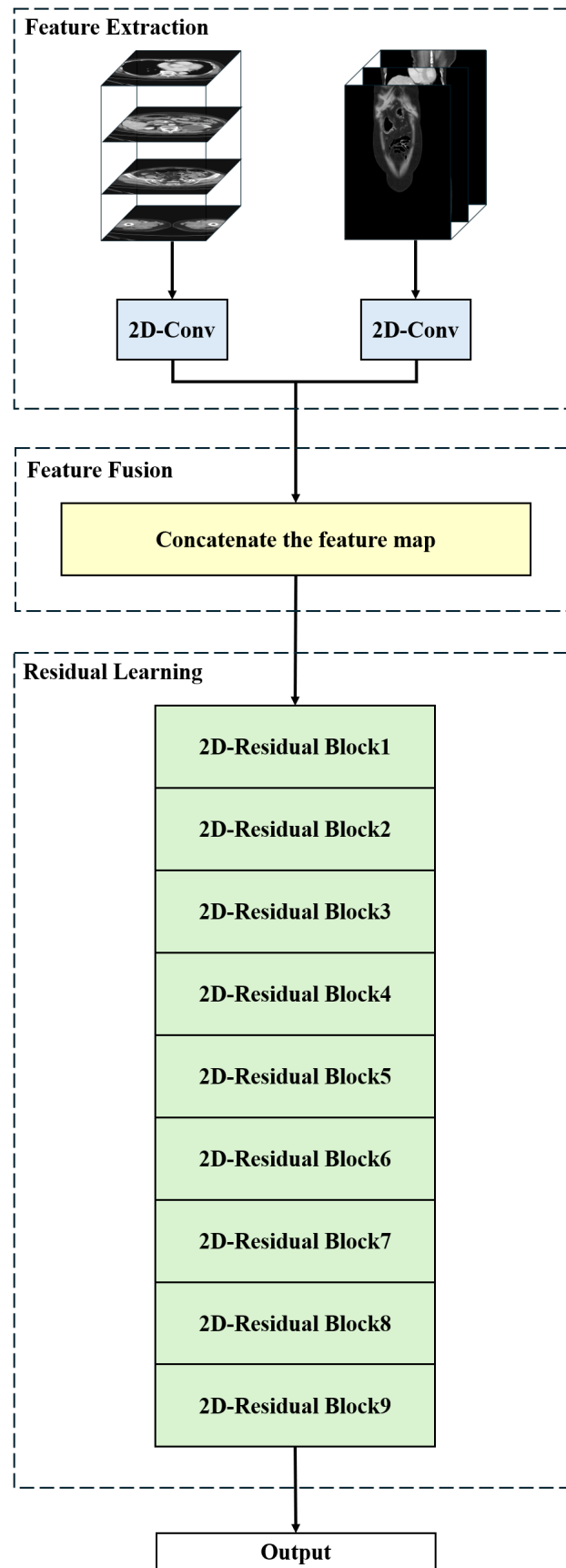


Figure 2.5: The overall architecture of the proposed deep learning model, highlighting the feature extraction, fusion, and residual learning stages.

2.3.1 Feature Extraction



In the feature extraction stage, the model receives processed axial and coronal thick-slice CT image sets. Each set typically has a size of $(10, 512, 512)$, where 10 represents the number of consecutive slices processed together, and 512×512 is the resolution of each slice. The model processes the axial-derived and coronal-derived image sets independently through separate 2D convolution pathways, as shown in Fig. 2.6. Each pathway initially employs a 3×3 convolution layer. These convolution operations are designed to efficiently capture low-level local texture information and features from their respective orientations, which form the fundamental representations for subsequent fusion and learning.

To improve feature stability and training speed, Batch Normalization (BN) [19] is applied to the convolved feature maps. BN normalizes the outputs of the convolution layers, reducing internal covariate shift and leading to more stable and faster training. Following BN, a Rectified Linear Unit (ReLU) [20] activation function is applied. ReLU introduces non-linearity into the model, allowing it to learn more complex patterns and better capture variations in image texture.

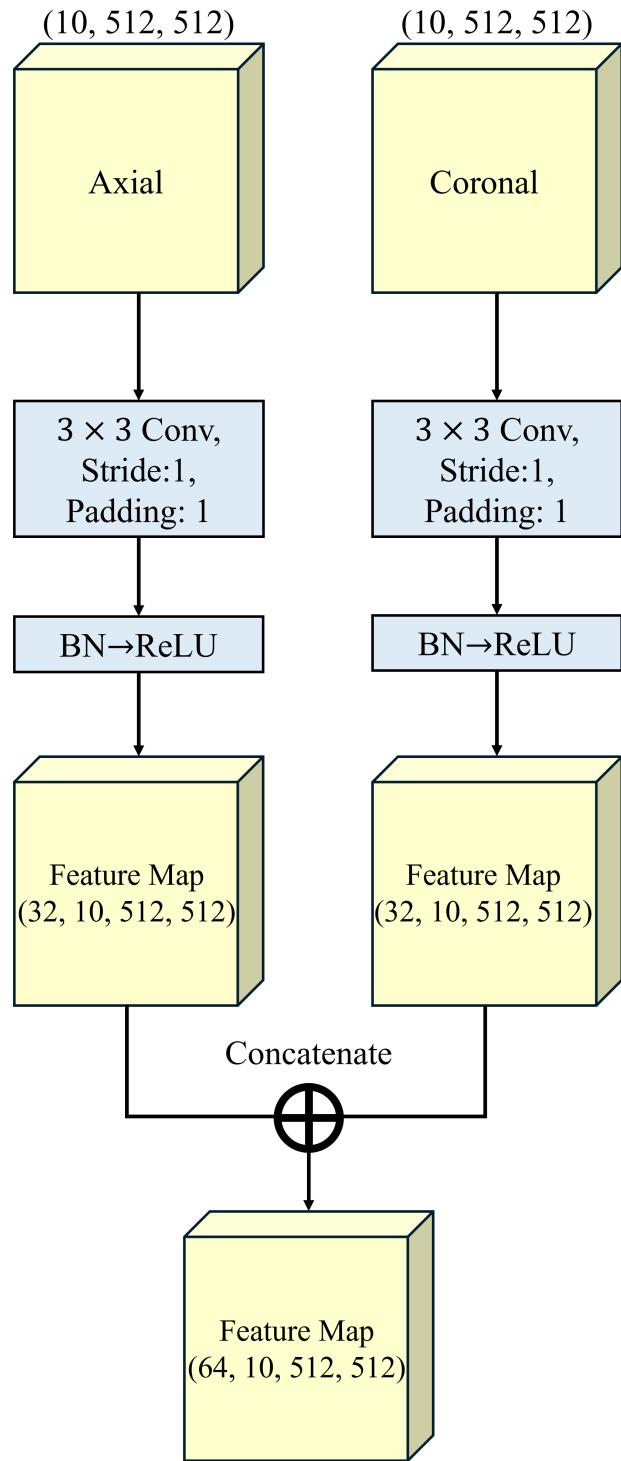


Figure 2.6: The detailed structure of the Feature Extraction Stage. Axial and coronal inputs are processed through separate convolutional pathways before feature map concatenation.

2.3.2 Feature Fusion



Recognizing that images from different orientations (axial and coronal) contain complementary information, the feature maps extracted from the separate axial and coronal pathways in the Feature Extraction Stage are then concatenated. This fusion step, also illustrated in Fig. 2.6, combines the learned representations. The resulting concatenated feature map has an increased channel dimension, thereby preserving critical information from multiple perspectives. This richer, fused feature map is then passed on to the subsequent Residual Learning Stage.

2.3.3 Residual Learning

In the residual learning [17] stage, the model employs a series of nine Residual Blocks to deeply learn and refine the image details from the fused features. A Residual Block, detailed in Fig. 2.7, is typically composed of two 3×3 convolution layers. Crucially, it includes a "skip connection" that adds the input of the block to its output. This architecture allows gradients to backpropagate more effectively through deep networks, mitigating the vanishing gradient problem and improving the representation capability of the network. The residual learning process enables the network to focus on learning the intricate differences between its current input and the desired target features, which is highly beneficial for restoring fine details, edges, and textures without losing information from earlier layers. After processing through all the residual blocks, the features are passed to a final output block to reconstruct the high-resolution thin-slice image.

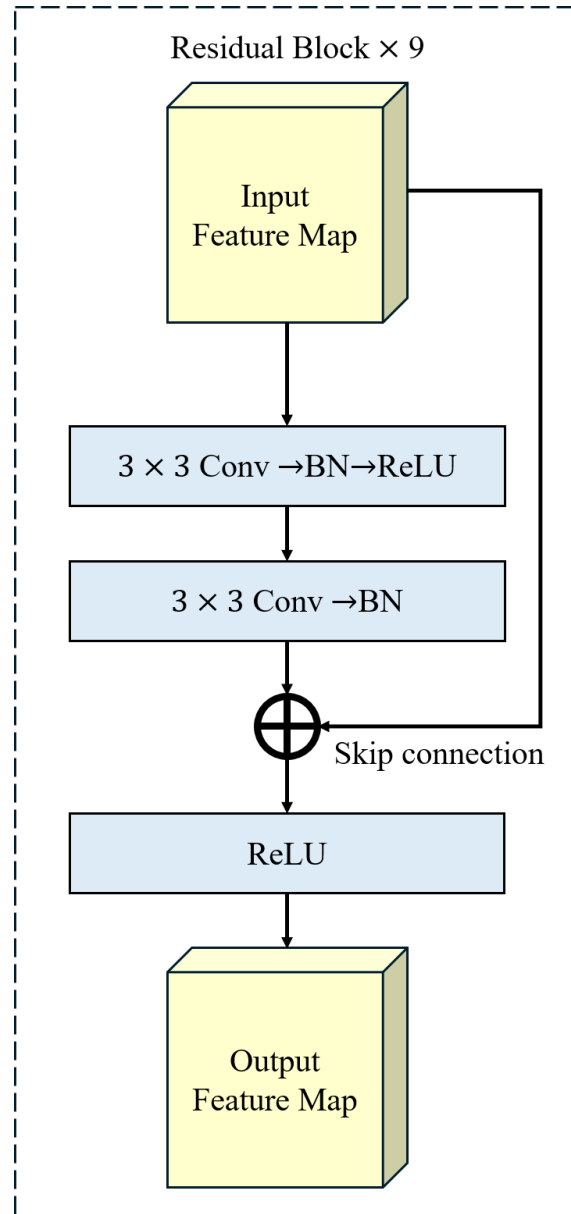


Figure 2.7: The detailed structure of a single residual block used in the Residual Learning Stage.

2.3.4 Loss Function

To guide the model training and ensure the quality of the restored images, this study utilizes a composite loss function designed for supervised learning. This loss function comprises three components: L1 Loss, L2 Loss, a 2D SSIM-based Loss, and a 3D SSIM-based Loss. The combination aims to restore texture, contrast, and fine details in the reconstructed images, ensuring they map closely to the ground truth.

The L1 Loss, or Mean Absolute Error (MAE), is defined as:

$$L_{L1} = \frac{1}{N} \sum_{i=1}^N |I_{\text{pred}}(i) - I_{\text{gt}}(i)| \quad (2.2)$$

where $I_{\text{pred}}(i)$ and $I_{\text{gt}}(i)$ denote the predicted and ground truth pixel values, respectively, and N is the total number of pixels.

The L2 Loss, or Mean SquaredError (MSE), is given by:

$$L_{L2} = \frac{1}{N} \sum_{i=1}^N (I_{\text{pred}}(i) - I_{\text{gt}}(i))^2 \quad (2.3)$$

The 2D Structural Similarity Index (2D SSIM) [21] is calculated based on three components: luminance (l), contrast (c), and structural similarity (s):

$$SSIM_{2D}(I_{\text{pred}}, I_{\text{gt}}) = l(I_{\text{pred}}, I_{\text{gt}}) \cdot c(I_{\text{pred}}, I_{\text{gt}}) \cdot s(I_{\text{pred}}, I_{\text{gt}}) \quad (2.4)$$

where:

$$l(I_{\text{pred}}, I_{\text{gt}}) = \frac{2\mu_{\text{pred}}\mu_{\text{gt}} + C_1}{\mu_{\text{pred}}^2 + \mu_{\text{gt}}^2 + C_1} \quad (2.5)$$

$$c(I_{\text{pred}}, I_{\text{gt}}) = \frac{2\sigma_{\text{pred}}\sigma_{\text{gt}} + C_2}{\sigma_{\text{pred}}^2 + \sigma_{\text{gt}}^2 + C_2} \quad (2.6)$$

$$s(I_{\text{pred}}, I_{\text{gt}}) = \frac{\sigma_{\text{pred,gt}} + C_3}{\sigma_{\text{pred}}\sigma_{\text{gt}} + C_3} \quad (2.7)$$

Here, μ and σ represent the mean and standard deviation of pixel values, respectively, and C_1, C_2, C_3 are small constants to stabilize the computation. To optimize for structural similarity, the 2D SSIM loss ($L_{\text{SSIM}_{2D}}$) is defined as:

$$L_{\text{SSIM}_{2D}} = 1 - \text{SSIM}_{2D}(I_{\text{pred}}, I_{\text{gt}}) \quad (2.8)$$

For volumetric data, the 3D Structural Similarity Index (3D SSIM) [22] extends the same principles as 2D SSIM, utilizing the same component functions for luminance, contrast, and structure:

$$\text{SSIM}_{3D}(V_{\text{pred}}, V_{\text{gt}}) = l(V_{\text{pred}}, V_{\text{gt}}) \cdot c(V_{\text{pred}}, V_{\text{gt}}) \cdot s(V_{\text{pred}}, V_{\text{gt}}) \quad (2.9)$$

The functions $l(\cdot)$, $c(\cdot)$, and $s(\cdot)$ are identical to those defined in Equations (2.5)–(2.7). The critical difference is that the local statistics are computed within a 3D volumetric window that moves through the volume.

The corresponding 3D SSIM loss ($L_{\text{SSIM}_{3D}}$) is defined as:

$$L_{\text{SSIM}_{3D}} = 1 - \text{SSIM}_{3D}(V_{\text{pred}}, V_{\text{gt}}) \quad (2.10)$$

The total loss function is a weighted sum of the aforementioned components, tailored to the specific model architecture.

For the 2D-CNN architecture, the total loss combines the L1, L2, and 2D SSIM losses:

$$L_{\text{total}} = \lambda_1 L_{\text{L1}} + \lambda_2 L_{\text{L2}} + \lambda_3 L_{\text{SSIM}_{2D}} \quad (2.11)$$

where λ_1 , λ_2 , and λ_3 are the respective weighting factors that balance the influence of each component.

For the 3D-CNN architecture, the total loss combines the L1, L2, and 3D SSIM losses:

$$L_{\text{total}} = \lambda_1 L_{L1} + \lambda_2 L_{L2} + \lambda_3 L_{\text{SSIM}_{3D}} \quad (2.12)$$

where λ_1 , λ_2 , and λ_3 are the corresponding weighting factors chosen for the 3D model training.

2.3.5 Training Settings

This study considered both 2D and 3D architectures for model training. An initial approach using a 3D-CNN, where 3D convolutions aimed to learn volumetric descriptors, was found to have very high computational costs and parameter counts. In practice, this model did not train efficiently due to limitations in available hardware, specifically insufficient GPU memory. Consequently, the primary focus shifted to a 2D-CNN architecture for image restoration. To ensure a fair basis for comparison between different architectural complexities, this study also involved gradually reducing the number of parameters in several 2D model variants, aiming to align their computational complexity more closely with that of the 3D model.

The Adam optimizer [23] was adopted to train the image reconstruction model, chosen for its advantages in self-adaptive learning rate adjustment and momentum, which contribute to improved convergence stability. The specific Adam optimizer settings are detailed in Table 2.1.

To control the learning rate during the training process and stabilize convergence in

Table 2.1: Hyperparameter settings for the Adam optimizer.

Hyperparameter	Value
Initial Learning Rate (η_0)	1×10^{-5}
Momentum Term 1 (β_1)	0.5
Momentum Term 2 (β_2)	0.999
Weight Decay	1×10^{-5}
Learning Rate Decay Factor (γ)	0.5
Step Interval (T)	5 epochs
Gradient Clipping Threshold	1.0



deeper networks, a StepLR scheduler was applied. This scheduler halves the learning rate every 5 epochs, according to the formula:

$$\eta_t = \eta_0 \times \gamma^{\lfloor t/T \rfloor} \quad (2.13)$$

where η_t is the learning rate at epoch t , η_0 is the initial learning rate (1×10^{-5}), γ is the decay factor (0.5), and T is the step interval (5 epochs).

Additionally, gradient clipping was applied with a maximum gradient norm constraint of 1.0. This prevents excessively large updates that can arise from sudden spikes in gradients, ensuring a more stable training process and facilitating faster convergence without sacrificing restoration accuracy.

The key hyperparameters used for training the models are summarized in Table 2.2.

Table 2.2: Training Hyperparameter details.

Hyperparameter	Value
Batch Size	2
Epochs	50
Optimizer	Adam
Initial Learning Rate	1×10^{-5}
Loss Function	L1 Loss + L2 Loss + SSIM Loss (weighted sum)

The model training and experiments were conducted in an NVIDIA GPU environment. Specific details of the training environment are provided in Table 2.3.



Table 2.3: Training Environment details.

Component	Specification
Python Version	3.9.17
Framework	PyTorch 2.4
Hardware	NVIDIA RTX 3090 (24GB VRAM)
Operating System	Windows 10
CUDA Version	12



Chapter 3 Results and Discussion

This chapter evaluates the proposed image restoration models, including the training results, quantitative evaluation metrics, and the impact of different model architectures on restoration quality.

3.1 Training Results

In this work, a variety of 2D and 3D image restoration models were trained, and the speed of convergence and computational cost were analyzed in this study, including one original 2D model (2D_Model_1), five parameter-reduced models (2D_Model_2 to 2D_Model_6), and one 3D convolutional model (3D_Model). We show the number of parameters and residual block channel configurations of each model in Table 3.1, while Table 3.2 presents the total training time. The training and validation loss curves are illustrated in Fig.3.1 through Fig.3.7.

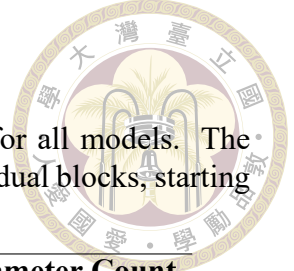
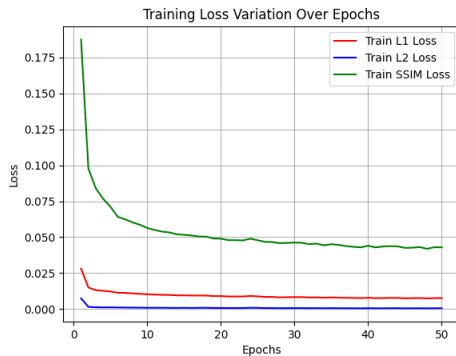


Table 3.1: Residual Block channel sequence and parameter count for all models. The sequence shows the number of output channels after each of the 9 residual blocks, starting from the initial input channels to Block 1.

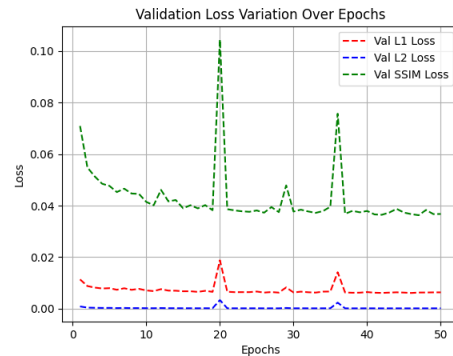
Model	Residual Block Channel Sequence	Parameter Count
2D_Model_1	$64 \rightarrow 128 \rightarrow 256^{\times 5} \rightarrow 128 \rightarrow 64 \rightarrow 10$	6,744,384
2D_Model_2	$48 \rightarrow 96 \rightarrow 192^{\times 5} \rightarrow 96 \rightarrow 48 \rightarrow 10$	3,798,240
2D_Model_3	$32 \rightarrow 64 \rightarrow 128^{\times 5} \rightarrow 64 \rightarrow 32 \rightarrow 10$	1,692,288
2D_Model_4	$24 \rightarrow 48 \rightarrow 96^{\times 5} \rightarrow 48 \rightarrow 24 \rightarrow 10$	954,384
2D_Model_5	$20 \rightarrow 32 \rightarrow 64^{\times 5} \rightarrow 32 \rightarrow 16 \rightarrow 10$	428,176
2D_Model_6	$20 \rightarrow 32^{\times 7} \rightarrow 16 \rightarrow 10$	145,616
3D_Model	$32 \rightarrow 16^{\times 8} \rightarrow 10$	114,840

Table 3.2: Training time and GPU memory cost for each model.

Model	Cost time(hr)	GPU Memory Cost(GB)
2D_Model_1	60.67	19
2D_Model_2	45	14.9
2D_Model_3	26.25	9.5
2D_Model_4	22.5	8
2D_Model_5	16.08	5.2
2D_Model_6	13.33	3.4
3D_Model	61.66	16.7

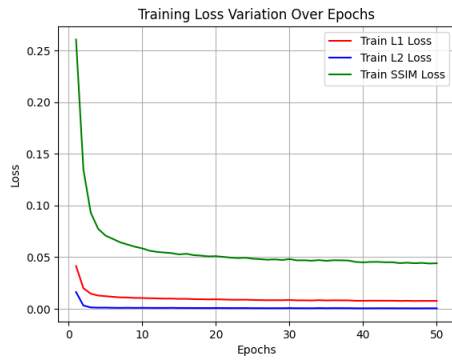


(a) Training Loss of 2D_Model_1

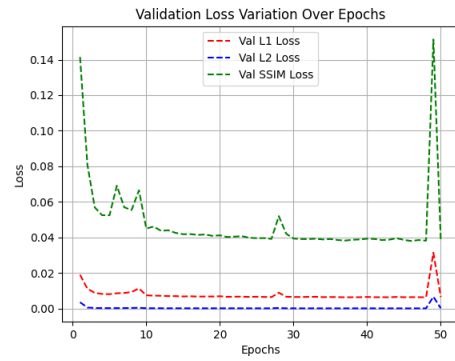


(b) Validation Loss of 2D_Model_1

Figure 3.1: Training and validation loss curves for 2D_Model_1.

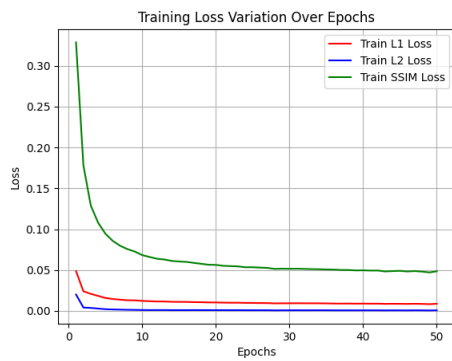


(a) Training Loss of 2D_Model_2

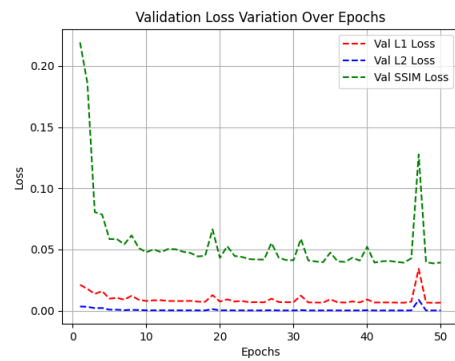


(b) Validation Loss of 2D_Model_2

Figure 3.2: Training and validation loss curves for 2D_Model_2.

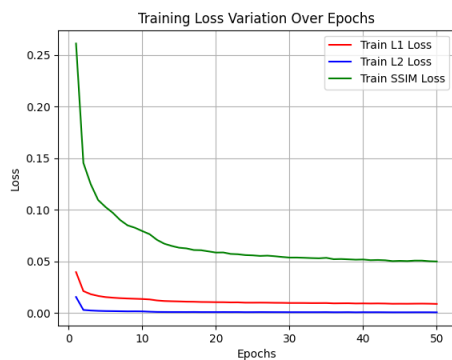


(a) Training Loss of 2D_Model_3

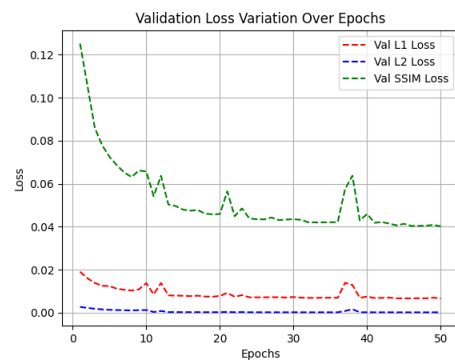


(b) Validation Loss of 2D_Model_3

Figure 3.3: Training and validation loss curves for 2D_Model_3.

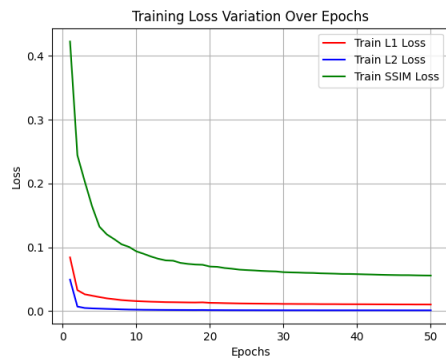
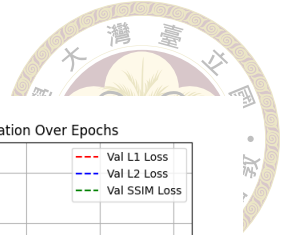


(a) Training Loss of 2D_Model_4

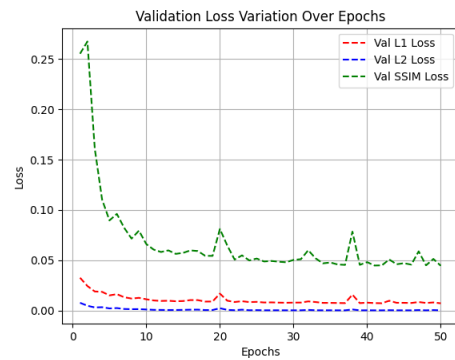


(b) Validation Loss of 2D_Model_4

Figure 3.4: Training and validation loss curves for 2D_Model_4.

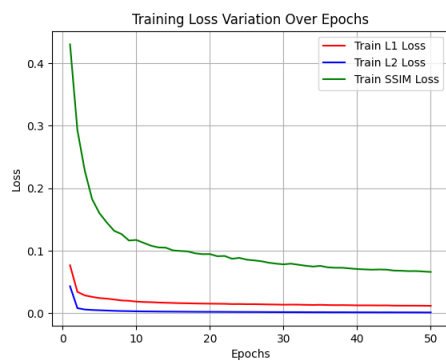


(a) Training Loss of 2D_Model_5

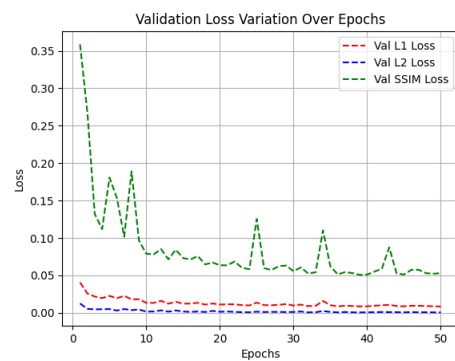


(b) Validation Loss of 2D_Model_5

Figure 3.5: Training and validation loss curves for 2D_Model_5.

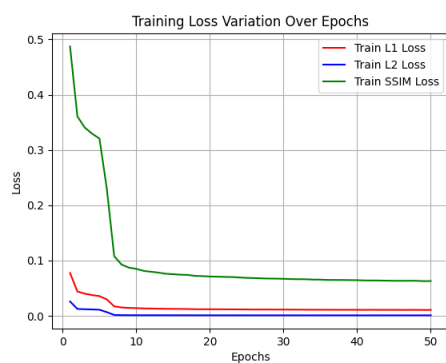


(a) Training Loss of 2D_Model_6

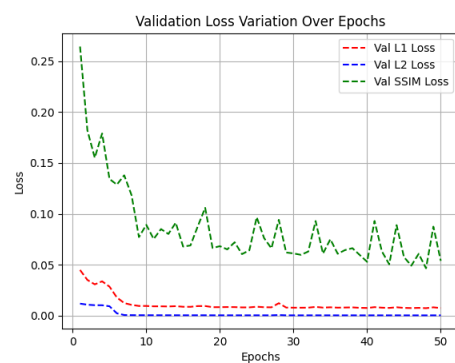


(b) Validation Loss of 2D_Model_6

Figure 3.6: Training and validation loss curves for 2D_Model_6.



(a) Training Loss of 3D_Model



(b) Validation Loss of 3D_Model

Figure 3.7: Training and validation loss curves for 3D_Model.

As shown in Fig.3.1 to Fig.3.7, all models show a steady decline in both training and validation losses, reflecting their ability to learn the non-linear mapping from thick-slice inputs to thin-slice target images to reconstruct images. Nonetheless, there are differences between the models regarding convergence speed, loss values, and training stability.

The best feature learning capability is found on 2D_Model_1 with most parameters. Its loss curves have a sharp drop in the initial stage, and the loss drops to a lower value and stabilizes in the following stage. It has the best convergence behavior and training stability. On the other hand, with a gradual reduction in the number of parameters from 2D_Model_2 to 2D_Model_6, we observe a steep decline in the convergence speed and relatively similar final loss values, indicating the balance between model complexity and its learning capability.

The mid sized models, like 2D_Model_2 and 2D_Model_3 show similar performance, both the training and validation losses are smoothly decreasing and are close to each other, indicating that they display stable learning behavior and some amount of generalisation. However, as the number of parameters decreases (as in 2D_Model_4, 2D_Model_5, and 2D_Model_6), a decline in training effectiveness and relatively higher validation loss are observed, which may indicate that the model capacity is insufficient to fully learn the complex features of the data.

Interestingly, the 3D_Model, despite having the fewest parameters among all models tested (Table 3.1), exhibited the least stable training behavior, characterized by significant initial loss variations and elevated validation loss, ultimately yielding the poorest performance. Its training time (61.66 hours) was notably long, nearly matching the most complex 2D_Model_1 (60.67 hours), despite its smaller size (Table 3.2). This highlights the

substantial computational overhead (time and memory consumption, 16.7 GB) associated with 3D convolutions. Furthermore, its weak performance suggests that, for this specific task of restoring thin slices from relatively thick inputs, the volumetric processing of 3D CNNs might be less effective than the proposed 2D dual-view approach, perhaps because the latter already captures sufficient cross-plane information through feature fusion, or because the limited available data and GPU resources hindered the effective training of the 3D architecture. The apparent lack of generalization ability further supports the choice of the 2D framework in this study.

3.2 Model Performance

This study adopts the SSIM as the primary assessment metric to quantitatively evaluate the performance of image restoration. SSIM values in nine test datasets are presented in Table 3.3.

Table 3.3: SSIM values of each model across different test data.

Data	2D_M1	2D_M2	2D_M3	2D_M4	2D_M5	2D_M6	3D
Data1	0.9501	0.9471	0.9469	0.9446	0.9411	0.9336	0.9322
Data2	0.9601	0.9589	0.9587	0.9587	0.9536	0.9448	0.9503
Data3	0.9736	0.9718	0.9712	0.9704	0.9644	0.9571	0.9613
Data4	0.9772	0.9757	0.9752	0.9747	0.9697	0.9615	0.9649
Data5	0.9515	0.9489	0.9485	0.9466	0.9424	0.9360	0.9369
Data6	0.9612	0.9600	0.9597	0.9602	0.9549	0.9437	0.9516
Data7	0.9601	0.9600	0.9570	0.9555	0.9510	0.9435	0.9469
Data8	0.9598	0.9596	0.9559	0.9544	0.9500	0.9413	0.9455
Data9	0.9551	0.9548	0.9507	0.9488	0.9453	0.9357	0.9393

As observed, 2D_Model_1 gives the highest SSIM score out of all the models which confirms that the more parameters in the network, the more complex the model and the better the structural preservation in the reconstructed image. The models 2D_Model_2 and 2D_Model_3 also achieve good performance with SSIM values being lower than

2D_Model_1 but overall with stable results over all datasets. These models provide a reasonable trade-off between the quality of reconstruction and the efficiency of the model. On the other hand, 2D_Model_5 and 2D_Model_6 exhibit gradual performance deterioration, with relatively low SSIM values specifically for Data3, Data4, and Data8. This indicates that excessive reduction in model capacity may lead to a loss of structural detail. Interestingly, SSIM score for 3D_Model is comparable to the 2D_Model_6 with a lesser number of parameters in some cases. Its performance is still a little worse than the majority of 2D varieties may be due to difficulty of optimization or lack of depth. The figures show comparisons of reconstructed results for all models, enabling inspection on the structural similarity and perceptual quality.

As a baseline for comparison, a few traditional interpolation methods were also tested, such as the Nearest Neighbor, Linear, Quadratic, and Cubic interpolation methods. Table 3.4 summarizes the average SSIM values for these methods on nine testing data.

Table 3.4: SSIM values of traditional interpolation methods across test datasets.

Data	Nearest Neighbor	Linear	Quadratic	Cubic
Data1	0.9324	0.9330	0.9378	0.9376
Data2	0.9508	0.9518	0.9550	0.9544
Data3	0.9599	0.9608	0.9639	0.9632
Data4	0.9659	0.9667	0.9693	0.9687
Data5	0.9362	0.9372	0.9418	0.9416
Data6	0.9525	0.9539	0.9569	0.9564
Data7	0.9435	0.9448	0.9499	0.9500
Data8	0.9416	0.9424	0.9474	0.9476
Data9	0.9369	0.9375	0.9428	0.9431

In all cases, the higher ordinal methods of interpolation such as Quadratic and Cubic performed better than lower order methods like Nearest Neighbor as seen in SSIM scores. Quadratic and Cubic interpolation occasionally returned SSIM scores similar to a few lightweight deep learning models. Nonetheless, deeper learning models like 2D_Model_1

consistently performed better than interpolation-based methods in each evaluation metric, indicating their superior capacity for structural detail and texture preservation through robust data-driven learning.



3.3 Model Comparison

In this section, comparisons of the restored thin-slice CT images are given to further evaluate the visual quality and restoration quality of the proposed models. Restoration outcomes tend to vary depending on the complexity of the model being used. Models like 2D_Model_1, which have higher SSIM values, are expected to retain structural information better and effectively extract meaningful features from both axial and coronal views. As illustrated in Fig. 3.8, all results are displayed using a window width (WW) of 400 and a window center (WC) of 40 for comparison, where each row presents the input images, restored outputs, and ground truth, displaying five consecutive slices to continuity between the slices. One can see from the red marked area which shows the model utilizes the complementary information from axial and coronal direction to restore the fine structural details with the degree of accuracy.

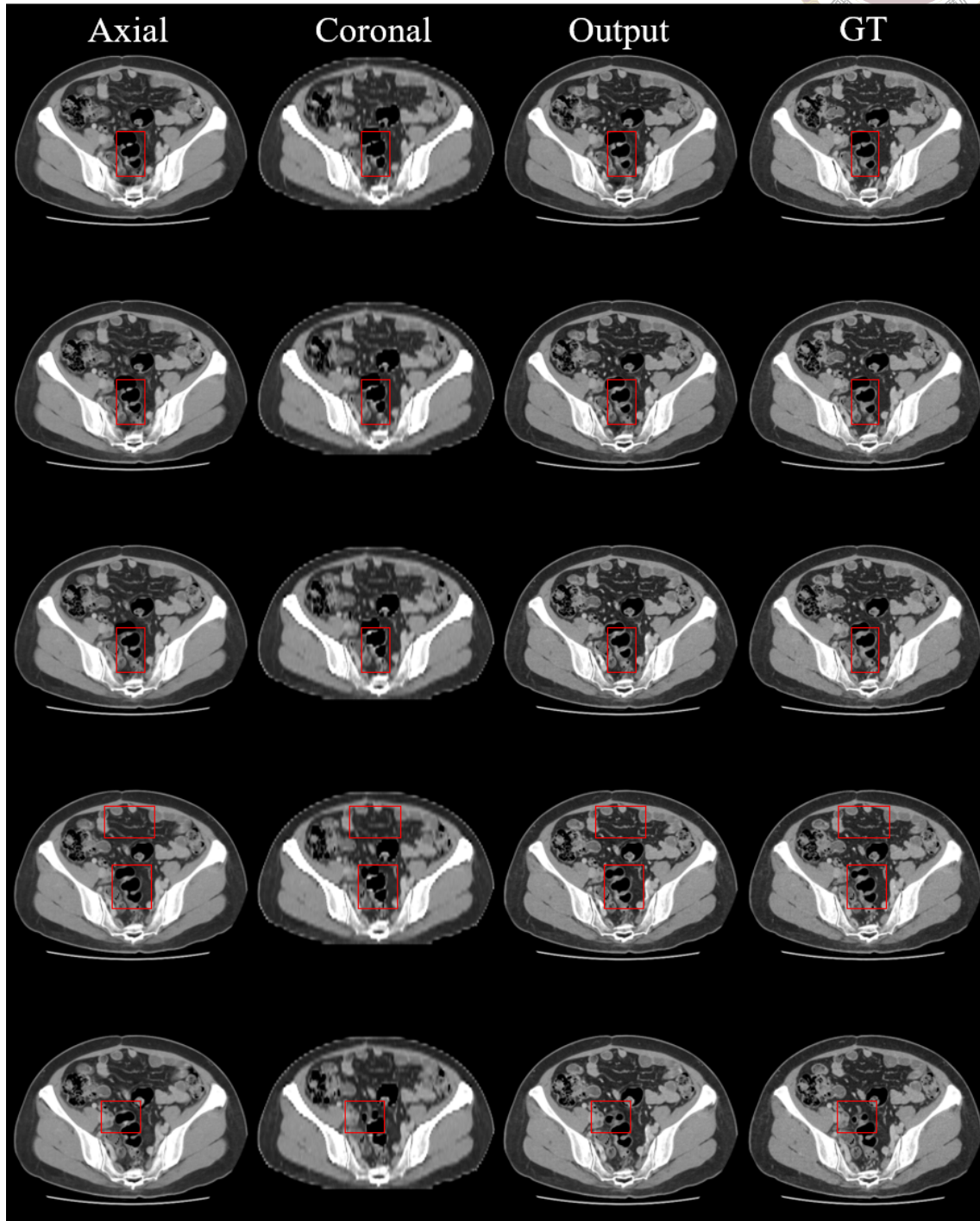


Figure 3.8: Restoration results of 2D_Model_1 showing five consecutive slices, including axial/coronal inputs, model output, and ground truth.

As demonstrated in Fig. 3.9 and Fig. 3.10, with this selected window width and window center, some anatomical structures highly visible in the ground truth images cannot be identified in the provided axial/coronal views.



Remarkably, despite certain anatomical structures being barely visible in the input thick-slice views under the selected display settings (WW/WC 400/40), the trained models, particularly those with higher capacity like 2D_Model_1, learned to restore these areas effectively. This ability to infer details suggests that the models are not merely interpolating but are actively learning complex spatial and contextual relationships from the training data. They appear capable of augmenting signals and deducing plausible pixel intensities that approximate the original HU values, likely by leveraging learned anatomical priors and texture patterns from surrounding visible structures. This capability highlights a key advantage of deep learning over traditional interpolation methods for this restoration task.

In the case of lightweight models, it may be blurred or missing texture details, and the further the number of model parameters decreases, the more obvious the phenomenon occurs around the edge. While the generated visual outputs of higher-capacity models like 2D_Model_1 and 2D_Model_2 are largely similar and still preserve detailed structures well, the reconstructions generated from smaller models become increasingly smoother and less texture-rich.

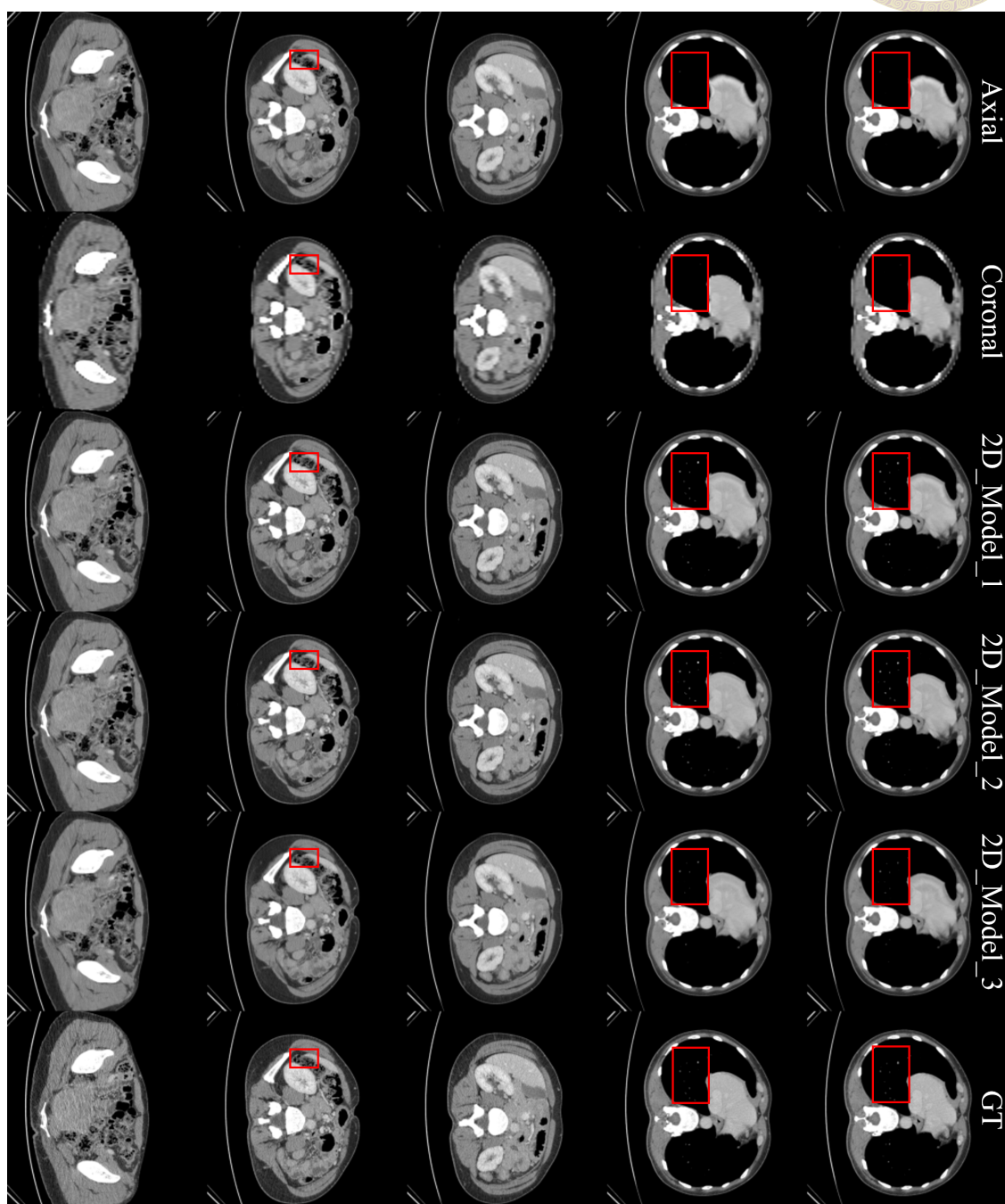


Figure 3.9: Comparison of restored images across 2D_Model_1 to 2D_Model_3.

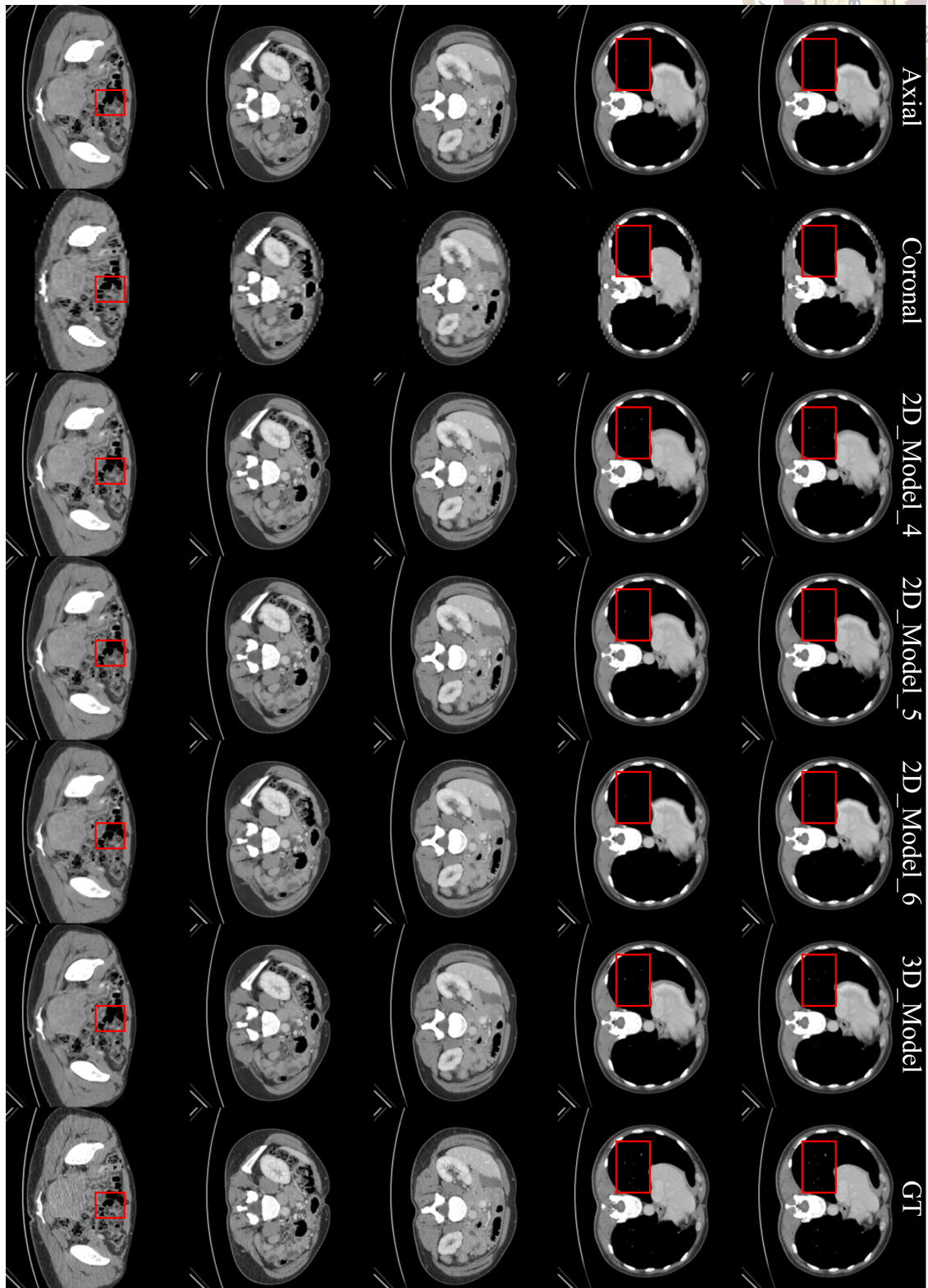
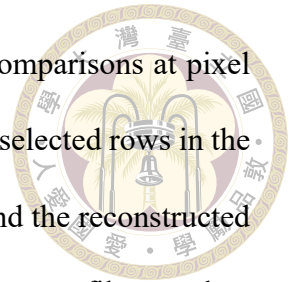


Figure 3.10: Comparison of restored images across 2D_Model_4 to 3D_Model.

Similar to the quantitative analyses provided above, intensity comparisons at pixel level were conducted as well. The horizontal intensity profiles of the selected rows in the axial and coronal planes were extracted from both the ground truth and the reconstructed image. Upon further study of the pixel profiles of the two images, these profiles are then displayed side-by-side in such a manner as to find subtle differences that exist between the two images in terms of pixel values. By plotting the intensity values side by side, it becomes easier to assess the consistency, sharpness, and local deviations of the reconstructed output with respect to the ground truth. This analysis identifies where smoothing, edge degradation, and intensity mismatches may occur in the restoration image. As the results using 2D_Model_1 are illustrated in Fig. 3.11 below. A sequence of the restored image, GT, and intensity profile from left to right. In the intensity profile, the y-axis represents pixel intensity (ranging from 0 to 255), and the x-axis represents image columns. The red dashed line is used to denote the column that we will focus on. From these plots, we see that the image boundaries stable, and the reconstruction is overall smoother and continuous.



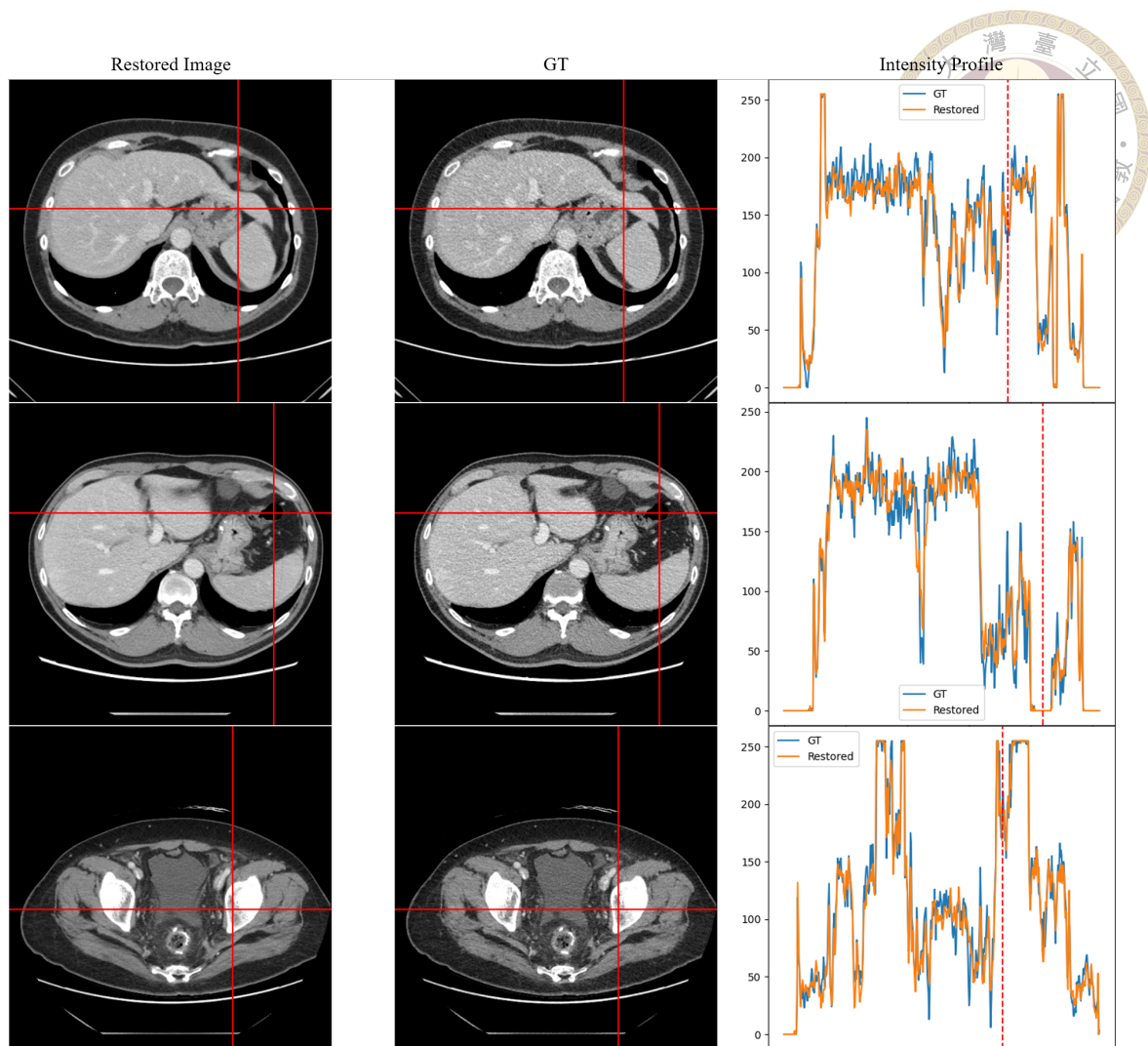


Figure 3.11: Pixel intensity profile using 2D_Model_1.

However, there will be instances when you might see some artifacts in restored images. Such artifacts often manifest as unnatural intensity variations or structural deformations, especially at the boundaries of complex anatomy or in areas of low signal contrast. These problems might occur when the model is not able to generalize to such highly varying textures. An example is shown in Fig. 3.12.

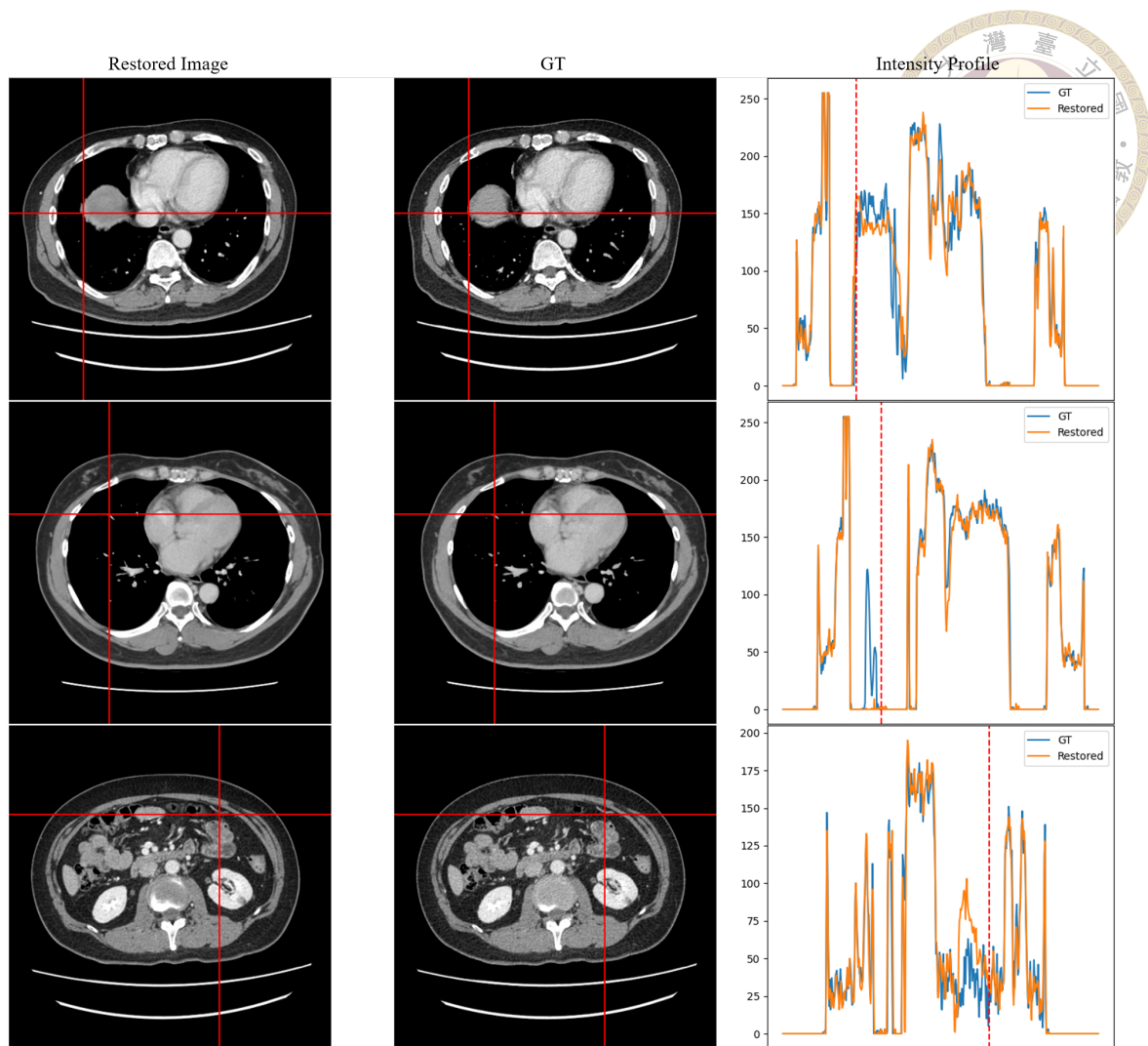


Figure 3.12: Example of pixel intensity profile showing artifacts in the restored image generated by 2D_Model_1.

Then, the pixel intensity profile is also presented in coronal plane according to the restored result. As shown in Fig. 3.13, some minor artifacts can still be seen in certain regions, while the overall performance at anatomical boundaries is very good. The restoration is smooth and consistent over the profile.

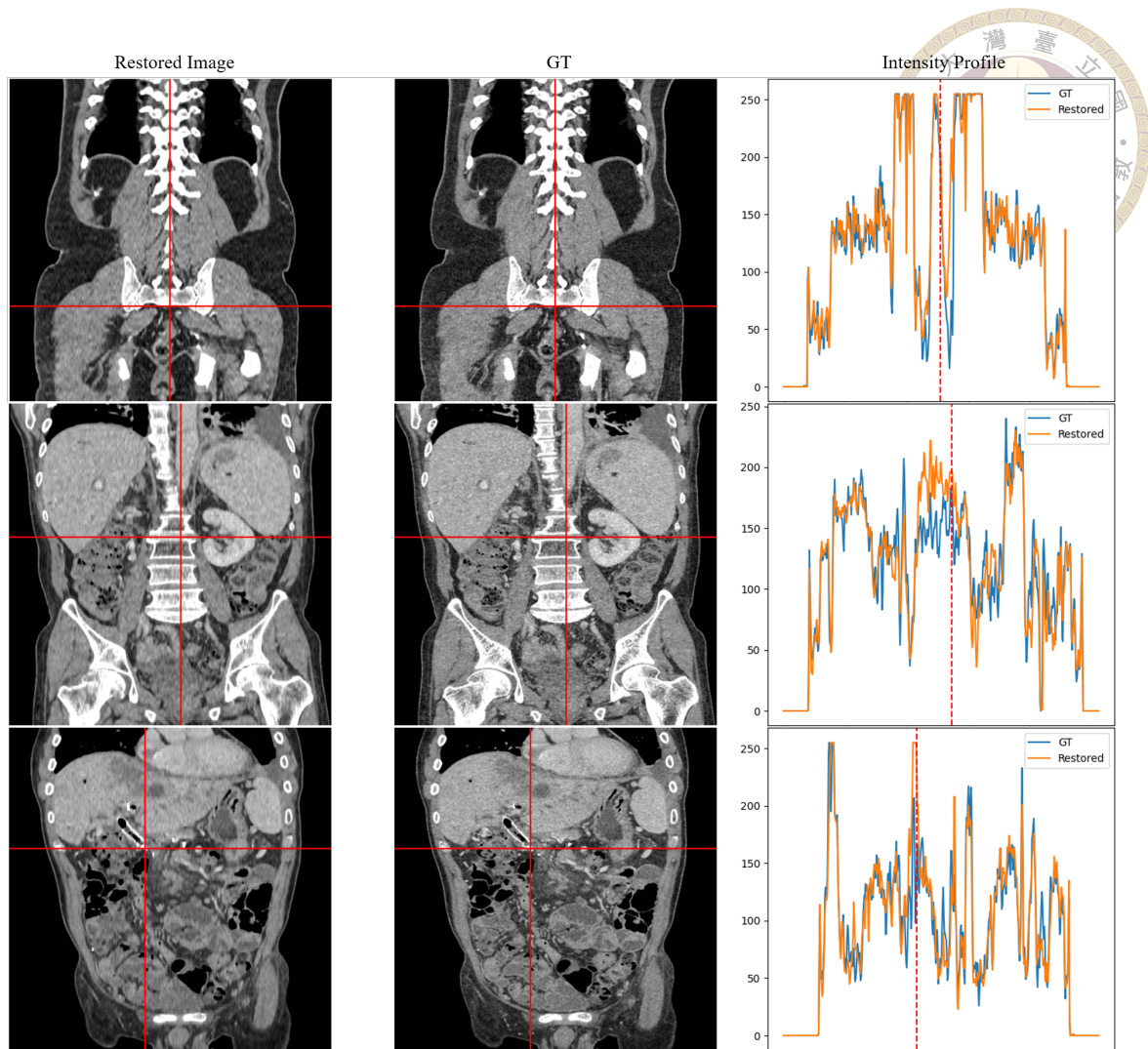


Figure 3.13: Pixel intensity profile along a selected row in the coronal plane using 2D_Model_1.

3.4 Performance Comparison of Single-View and Dual-View Models

To further assess the contribution of individual input views and validate the advantages of the proposed dual-view fusion strategy, this study conducted a comparative analysis between single-view models and the original dual-view model (2D_Model_1). The single-view models were derived from the 2D_Model_1 architecture by retaining only the corresponding single input branch and removing the subsequent feature concatenation layer. All other network layers, architectural depth, and training hyperparameters were

kept identical to ensure a fair comparison. The SSIM values for these models across the nine test data are detailed in Table 3.5.

Table 3.5: Comparison of SSIM values for single-view models and the 2D_Model_1 across test datasets.

Test Dataset	Only coronal data	Only axial data	2D_Model_1
Data1	0.7077	0.9413	0.9501
Data2	0.7253	0.9543	0.9601
Data3	0.7515	0.9636	0.9736
Data4	0.7781	0.9564	0.9772
Data5	0.7097	0.9442	0.9515
Data6	0.7124	0.9563	0.9612
Data7	0.6970	0.9244	0.9601
Data8	0.7087	0.9424	0.9598
Data9	0.7076	0.9414	0.9551
Average SSIM	0.7220	0.9472	0.9610

As shown in Table 3.5, the model using only coronally-reconstructed axial inputs produced a notably lower average SSIM of 0.7220. This suggests that relying on this single view alone provides insufficient information for high-fidelity restoration of axial slices. In contrast, the axial single-view model performed much better, with an average SSIM of 0.9472. However, the dual-view model (2D_Model_1) still achieved the highest average SSIM of 0.9610 and outperformed the others on every test dataset. Figure 3.14 visually corroborates these quantitative findings. It presents a comparison of representative restored slices from the ground truth (GT), the dual-view model (2D_Model_1), the axial-only single-view model, and the coronal-only single-view model. From the figure, it is evident that 2D_Model_1 achieves superior restoration of fine anatomical details and structural compared to both single-view approaches, which tend to exhibit more blurring or lose finer details. These comparative analysis explicitly underscores the superiority and necessity of the dual-view input strategy for the thin-slice CT image restoration task investigated in this study.

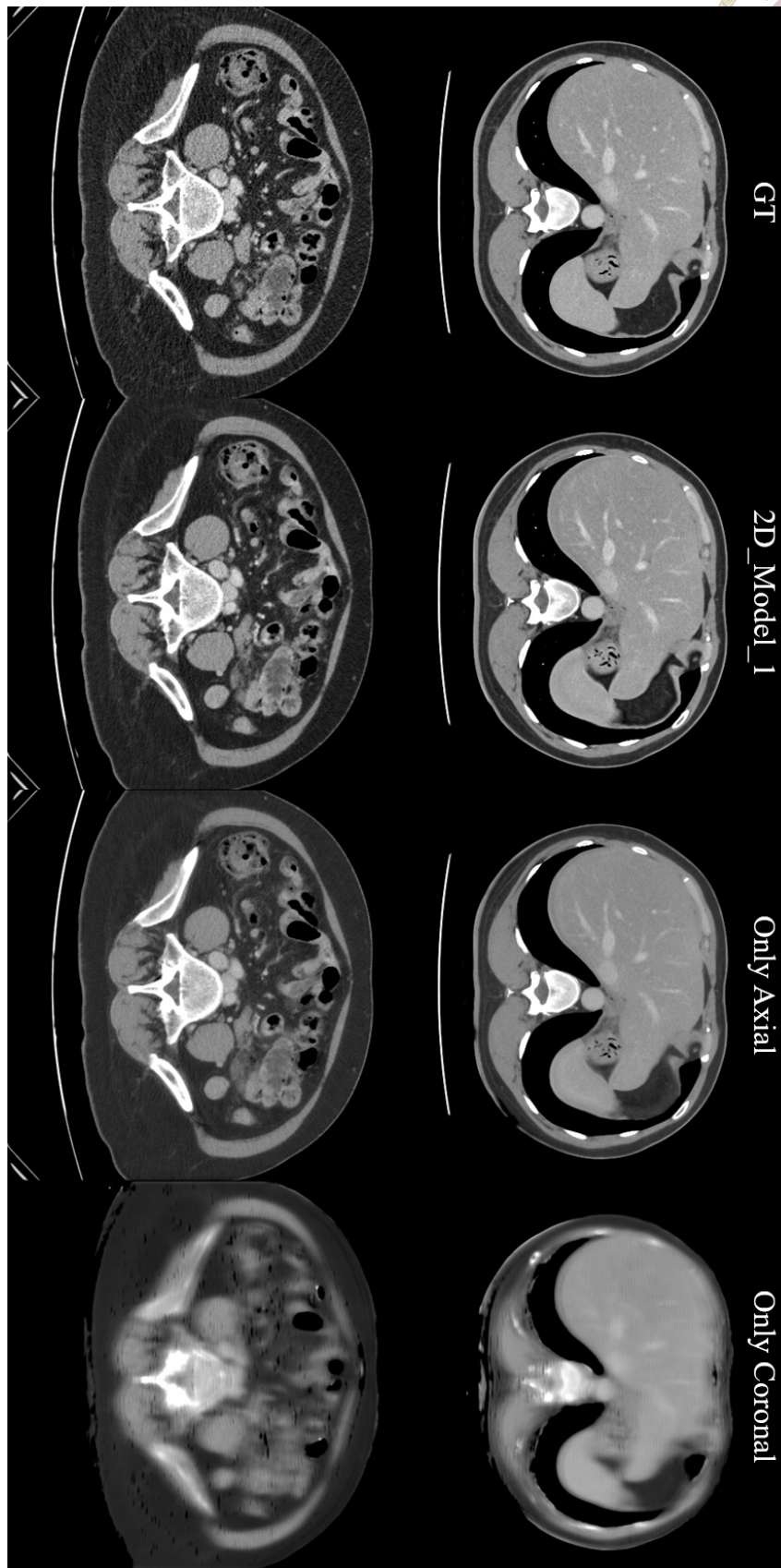
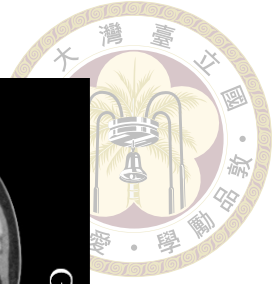


Figure 3.14: Visual comparison of thin-slice CT image restoration. The dual-view approach demonstrates superior detail recovery and structural integrity.

3.5 Simulation



In order to observe how the proposed method performs when it is subjected to different input conditions, we ran a simulation experiment with the best model, 2D_Model_1. The simulation methodology, which involves generating 10mm, 15mm, or 20mm thick inputs from 1mm ground truth data and subsequently using 2D_Model_1 to restore these inputs to 1mm resolution, is depicted in Figure 3.15. For this experiment, the aforementioned simulated inputs were generated from ground truth 1mm CT images taken from the test set. Specifically, these simulated inputs were produced by averaging every n sequential 1mm slices (thicknesses of 10mm, 15mm, and 20mm for $n = 10, 15$, and 20 , respectively) in order to test the n -fold data compression ratios from which the model could successfully restore the original 1mm resolution. The original 1mm images also served as the ground truth for evaluating the model's restoration performance.

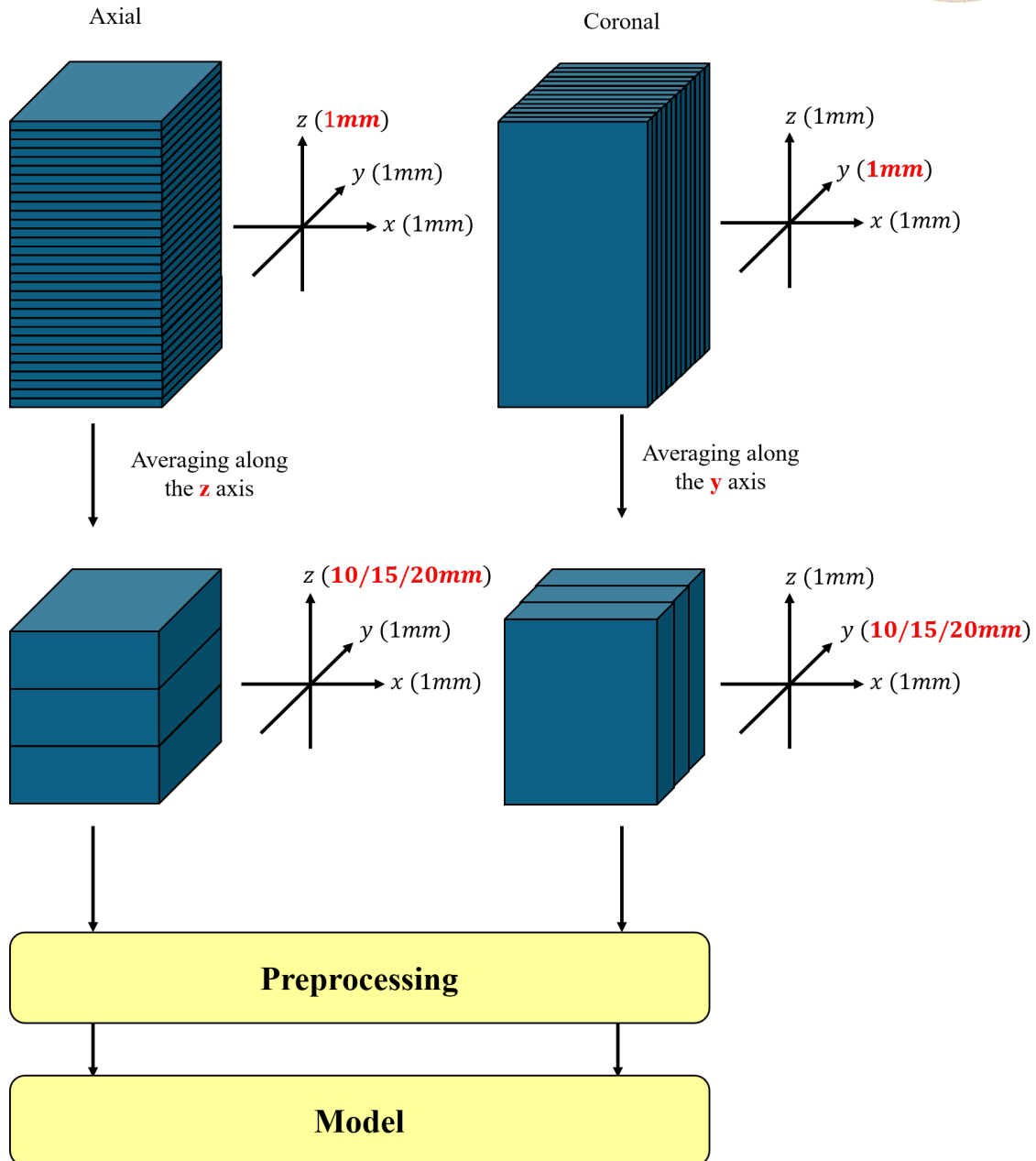


Figure 3.15: Illustration of the simulation pipeline: 1mm ground truth (GT) images are averaged to create 10mm, 15mm, or 20mm thick simulated inputs. These inputs are then processed by 2D_Model_1 to produce a restored 1mm resolution output.

Using these simulated inputs, we applied 2D_Model_1 to reproduce all nine test cases (Data1 to Data9). The SSIM results are shown in Table 3.6. For 10 mm input, and maximum SSIM, the model attained an average SSIM of 0.9074 and therefore good reconstruction capabilities for moderately downsampled data.

Table 3.6: SSIM performance of simulated inputs restored by 2D_Model_1

Data	10mm SSIM	15mm SSIM	20mm SSIM
Data1	0.8931	0.8645	0.8503
Data2	0.9258	0.9049	0.8894
Data3	0.9234	0.8954	0.8834
Data4	0.9338	0.9106	0.8991
Data5	0.8997	0.8738	0.8563
Data6	0.9290	0.9076	0.8938
Data7	0.8973	0.8724	0.8556
Data8	0.8955	0.8685	0.8583
Data9	0.9040	0.8804	0.8646
Average	0.9074	0.8864	0.8722

However, as the simulated input slice thickness increased to 15mm and 20mm, the average SSIM dropped significantly to 0.8864 and 0.8722, respectively. This performance degradation likely stems from two primary factors exacerbated by the thicker slices. First, the averaging process inherently causes a substantial loss of high-frequency details and sharp edges present in the original 1mm data, information crucial for high-fidelity reconstruction. Second, the increased slice thickness leads to more severe partial volume averaging artifacts, potentially introducing ambiguous signals or blurring boundaries in a way that confuses the model, which was primarily trained on 5mm to 1mm restoration. Reconstructing high-fidelity images from such heavily degraded inputs thus remains a significant challenge. Figure 3.16 illustrates this trend, comparing the Ground Truth with the outputs of 2D_Model_1 for inputs of 10mm, 15mm, and 20mm thickness. A clear decline in image quality, such as increased blurring and loss of fine structures, can be observed as the input slice thickness increases.

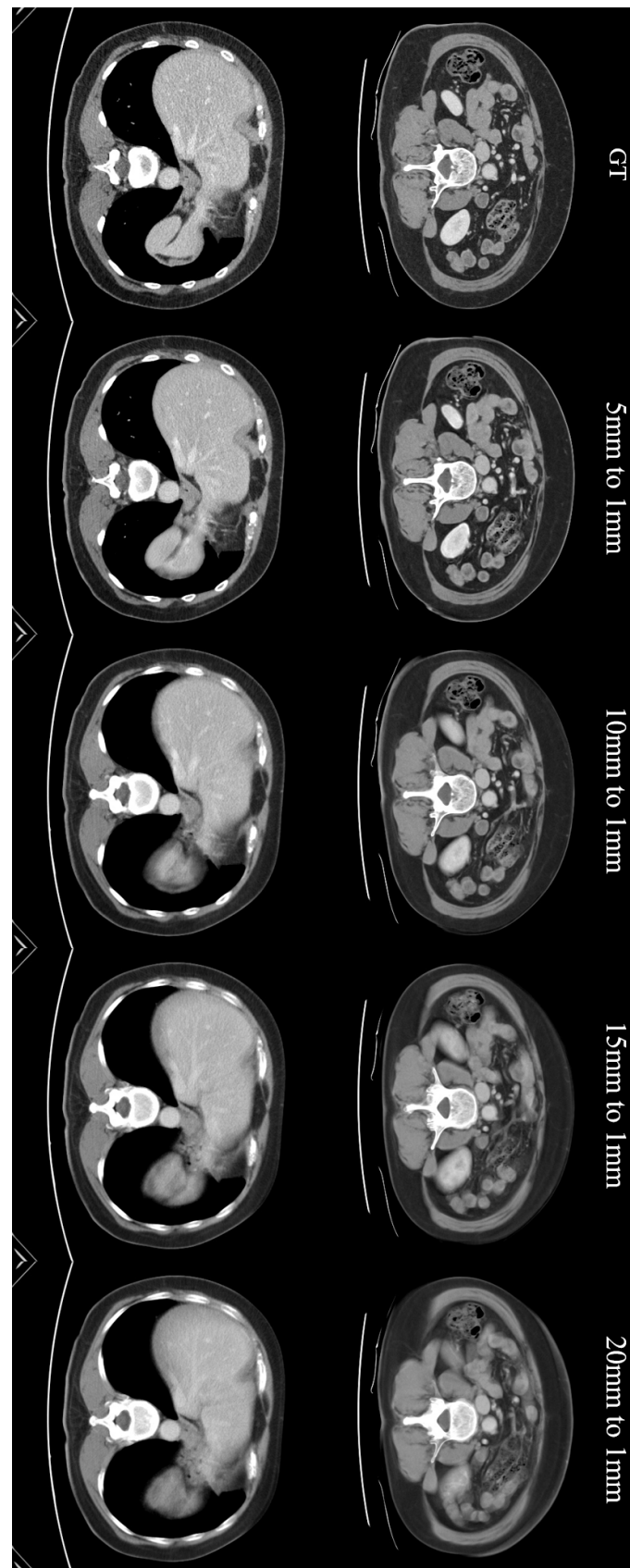


Figure 3.16: Visual comparison of 2D_Model_1 restoration performance on simulated thick-slice inputs.

Notably, Data4 outperformed all other datasets at all thickness levels, while Data1 performed the worst, indicating that anatomical differences between samples can impact model sensitivity. This simulation also explored the model's efficiency for data compression. The success of 2D_Model_1 in restoring 1mm resolution from 10mm inputs demonstrates a significant potential in this regard. Specifically, storing data as 10mm slices instead of 1mm offers an approximate 10-fold reduction in data volume, with the model enabling to recovery of high-resolution details.

3.6 Study Limitations

Despite the encouraging performance, this study has limitations that open avenues for future research. The GPU memory constraints encountered during training may have limited the exploration of deeper or more complex architectures. Furthermore, while using real clinical data, the dataset diversity could be expanded to ensure broader generalization across different patient populations, scanner types, and clinical scenarios. The dual-view fusion strategy itself could also be further refined. To address these, future work could focus on: (1) Incorporating advanced network modules, such as attention mechanisms [24, 25] or transformer [26] architectures, specifically designed to better capture long-range spatial dependencies and potentially improve fusion efficiency; and (2) Training and validating models on larger, multi-center and potentially multi-modality datasets to enhance robustness and clinical applicability



Chapter 4 Conclusion

We have implemented a dual-view 2D CNN model to restore the spatial resolution of CT images from existing thick-slice images. The model provided a better balance between restoration fidelity and computation complexity when compared with interpolation methods and the 3D CNN model with a comparable number of parameters. The restored images were finer and isotropic in spatial resolution, which is compatible with volume rendering useful for diagnosis or treatment evaluation. The tissue textures and edges were satisfactorily preserved with an image compression ratio up to 10, which helps data storage and transmission. Although the validation loss exhibited some fluctuations during training, the overall downward trend and strong performance on the test set indicate that this behavior might reflect benign overfitting [27]. Temporary adaptation to sample-specific features during training may lead to transient peaks in validation loss, without impairing the model's generalization capability. Further study with larger and multicenter datasets is worthwhile to improve the model generalization.



References

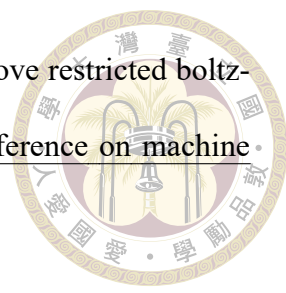
- [1] Luuk J. Oostveen, Ewoud J. Smit, Helena M. Dekker, Constantinus F. Buckens, Sjoert A. H. Pegge, Frank de Lange, Ioannis Sechopoulos, and Mathias Prokop. Abdominopelvic ct image quality: Evaluation of thin (0.5-mm) slices using deep learning reconstruction. American Journal of Roentgenology, 220(3):381–388, 2023. PMID: 36259592.
- [2] P. Thevenaz, T. Blu, and M. Unser. Interpolation revisited [medical images application]. IEEE Transactions on Medical Imaging, 19(7):739–758, 2000.
- [3] R. Lu, P. Marziliano, and C.H. Thng. Comparison of scene-based interpolation methods applied to ct abdominal images. In The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, volume 1, pages 1561–1564, 2004.
- [4] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. Nature, 521(7553):436–444, May 2015.
- [5] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A.W.M. van der Laak, Bram van Ginneken, and Clara I. Sánchez. A survey on deep learning in medical image analysis. Medical Image Analysis, 42:60–88, 2017.

- 
- [6] Sabina Umirzakova, Shabir Ahmad, Latif U. Khan, and Taegkeun Whangbo. Medical image super-resolution for smart healthcare applications: A comprehensive survey. Information Fusion, 103:102075, 2024.
- [7] Yanfei Jia, Guangda Chen, and Haotian Chi. Retinal fundus image super-resolution based on generative adversarial network guided with vascular structure prior. Scientific Reports, 14(1):22786, Oct 2024.
- [8] M. L. de Leeuw den Bouter, G. Ippolito, T. P. A. O'Reilly, R. F. Remis, M. B. van Gijzen, and A. G. Webb. Deep learning-based single image super-resolution for low-field mr brain images. Scientific Reports, 12(1):6362, Apr 2022.
- [9] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence, 38(2):295–307, 2015.
- [10] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1646–1654, 2016.
- [11] Yukai Wang, Qizhi Teng, Xiaohai He, Junxi Feng, and Tingrong Zhang. Ct-image of rock samples super resolution using 3d convolutional neural network. Computers Geosciences, 133:104314, 2019.
- [12] Li Kang, Bin Tang, Jianjun Huang, and Jianping Li. 3d-mri super-resolution reconstruction using multi-modality based on multi-resolution cnn. Computer Methods and Programs in Biomedicine, 248:108110, 2024.
- [13] Xianling Dong, Shiqi Xu, Yanli Liu, Aihui Wang, M. Iqbal Saripan, Li Li, Xiaolei

Zhang, and Lijun Lu. Multi-view secondary input collaborative deep learning for lung nodule 3d segmentation. Cancer Imaging, 20(1):53, Aug 2020.



- [14] Yang Qu, Xiaomin Li, Zhennan Yan, Liang Zhao, Lichi Zhang, Chang Liu, Shuaining Xie, Kang Li, Dimitris Metaxas, Wen Wu, Yongqiang Hao, Kerong Dai, Shaoting Zhang, Xiaofeng Tao, and Songtao Ai. Surgical planning of pelvic tumor using multi-view cnn with relation-context representation learning. Medical Image Analysis, 69:101954, 2021.
- [15] Lennart R Koetzier, Domenico Mastrodicasa, Timothy P Szczykutowicz, Niels R van der Werf, Adam S Wang, Veit Sandfort, Aart J van der Molen, Dominik Fleischmann, and Martin J Willemink. Deep learning image reconstruction for ct: technical principles and clinical prospects. Radiology, 306(3):e221257, 2023.
- [16] Heng-Sheng Chao, Yu-Hong Wu, Linda Siana, and Yuh-Min Chen. Generating high-resolution ct slices from two image series using deep-learning-based resolution enhancement methods. Diagnostics, 12(11), 2022.
- [17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.
- [18] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7132–7141, 2018.
- [19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

- 
- [20] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th international conference on machine learning (ICML-10), pages 807–814, 2010.
- [21] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4):600–612, 2004.
- [22] Kai Zeng and Zhou Wang. 3d-ssim for video quality assessment. In 2012 19th IEEE International Conference on Image Processing, pages 621–624, 2012.
- [23] Diederik P Kingma. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
- [24] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.
- [25] Jianning Chi, Zhiyi Sun, Huan Wang, Pengfei Lyu, Xiaosheng Yu, and Chengdong Wu. Ct image super-resolution reconstruction based on global hybrid attention. Computers in Biology and Medicine, 150:106112, 2022.
- [26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017.
- [27] Yuan Cao, Zixiang Chen, Misha Belkin, and Quanquan Gu. Benign overfitting in two-layer convolutional neural networks. In S. Koyejo, S. Mohamed, A. Agarwal,

D. Belgrave, K. Cho, and A. Oh, editors, Advances in Neural Information Processing Systems, volume 35, pages 25237–25250. Curran Associates, Inc., 2022.

