High-Resolution X-ray Image Generation from CT data using Super-Resolution

by

Qing Ma

Thesis submitted to the University of Ottawa
in partial fulfillment of the requirements for the
Master's degree in Computer Science

Ottawa-Carleton Institute for Computer Science
School of Electrical Engineering and Computer Science
University of Ottawa

Abstract

Synthetic X-ray or digitally reconstructed radiographs (DRRs) are simulated X-ray images projected from computed tomography (CT) data that are commonly used for CT and real X-Ray image registration. High-quality synthetic X-ray images can facilitate various applications such as guiding images for virtual reality (VR) simulation and training data for deep learning methods such as creating CT data from X-Ray images.

It is challenging to generate high-quality synthetic X-ray images from CT slices, especially in various view angles, due to gaps between CT slices, high computational cost, and the complexity of algorithms. Most synthetic X-ray generation methods use fast ray-tracing in a situation where the image quality demand is low. We aim to improve image quality while maintaining good accuracy and use two steps; 1) to generate synthetic X-ray images from CT data and 2) to increase the resolution of the synthetic X-ray images.

Our synthetic X-ray image generation method adopts a matrix-based projection method and dynamic multi-segment lookup tables, which shows better image quality and efficiency compared to conventional synthetic X-ray image generation methods. Our method is tested in a real-time VR training system for image-guided intervention procedures.

Then we proposed two novel approaches to raise the quality of synthetic X-ray images through deep learning methods. We use a reference-based super-resolution (RefSR) method as a base model to upsampling low-resolution images into higher resolution. Even though RefSR can produce fine details by utilizing the reference image, it inevitably generates some artifacts and noise. We propose texture transformer super-resolution with frequency domain (TTSR-FD) which introduces frequency domain loss as a constraint to improve the quality of the RefSR results with fine details and without apparent artifacts. To the best of our knowledge, this is the first work that utilizes frequency domain as a part of loss functions in the field of super-resolution (SR). We observe improved performance in evaluating TTSR-FD when tested on our synthetic X-ray and real X-ray image datasets.

A typical SR network is trained with paired high-resolution (HR) and low-resolution (LR) images, where LR images are created by downsampling HR images using a specific kernel. The same downsampling kernel is also used to create test LR images from HR images. As a result, most SR

methods only perform well when the testing image is acquired using the same downsampling kernel used during the training process. We also propose TTSR-DMK, which uses multiple downsampling kernels during training to generalize the model and adopt a dual model that trains together with the main model. The dual model can form a closed-loop with the main model to learn the inverse mapping, which further improves the model's performance. Our method works well for testing images produced by multiple kernels used during training. It can also help improve the model performance when testing images are acquired with kernels not used during training. To the best of our knowledge, we are the first to use the closed-loop method in RefSR.

We have achieved: (i) synthetic X-ray image generation from CT data, which is based on a matrix-based projection and lookup tables ; (ii) TTSR-FD: synthetic X-ray image super-resolution using a novel frequency domain loss ; (iii) TTSR-DMK: an adaptation network to overcome the performance drop for testing data which do not match to downsampling kernels used in training.

Our TTSR-FD results show improvements (PSNR from 37.953 to 39. 009) compared to the state-of-the-art methods TTSR. Our experiment with real X-Ray images using TTSR-FD can remove visible artifacts in the qualitative study even though PSNR is similar. Our proposed adaptation network, TTSR-DMK, improved model performance for multiple kernels even with unknown kernel situations.

Acknowledgment

I would like to express my deepest gratitude to my supervisor Professor WonSook Lee for her continued support and guidance throughout my research. She provided valuable and insightful suggestions on my research and offered me an excellent opportunity to work on innovative topics on medical imaging. I also want to thank Dr. Jae Chul Koh, who provided relevant data to this study and spent precious time collaborating with me remotely on the VR simulation project, which could potentially utilize my research in the future.

I am grateful to work in the LIII lab with my talented colleagues. The seminar and coffee time had enriched my master's study, especially during this unprecedented pandemic. The support I received from them both on academics and life is essential. I want to acknowledge and thank Jiawei Li for collaborating with me on the generation method during the initial period of my research.

I also want to thank my parents, grandparents, and girlfriend for their never-ending support and encouragement throughout my study. I could not achieve any of this without them.

Dedication

For my beloved parents Yanhua and Xijie, my lovely girlfriend Yunping. And for my grandpa Guoquan, who is gradually forgetting the whole world but remembered by me forever.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

AX	Axial view
CNN	Convolutional Neural Networks
COR	Coronal view
CT	Computed Tomography
DFT	Discrete Fourier transform
DICOM	Digital Imaging and Communications in Medicine
DRRs	Digital Reconstructed Radiographs
FFT	Fast Fourier transform
GAN	Generative Adversarial Networks
GT	Ground Truth
HR	High-Resolution image
HU	Hounsfield Units
IGT	Image-Guided Therapy
IQA	Image Quality Assessment
LR	Low-Resolution image.
MAE	Mean Absolute Error
MC	Monte Carlo simulation
MOS	Mean Opinion Score
MRI	Magnetic Resonance Imaging
MS	Magnitude Spectrums
NLP	Natural Language Processing

PIL	Python Imaging Library
PNG	Portable Network Graphics
PSNR	Peak Signal-to-Noise Ratio
Ref	Reference image
RefSR	Reference-based Super-Resolution
RNNs	Recurrent Neural Networks
SAG	Sagittal view
SISR	Single Image Super-Resolution
SR	Super-Resolution image
SSIM	Structural Similarity Index
VR	Virtual Reality

Chapter 1. Introduction

Image reconstruction or synthetic medical image generation has been an important technique in medical imaging. The demand for high-quality medical imaging data has increased with emerging technologies such as virtual reality (VR) training simulation and deep learning-based diagnosis. While medical data is always challenging to collect, synthetic images can mitigate this issue for particular application scenarios. In this study, we focus on producing high-quality synthetic X-ray images from CT data. Generating high-quality synthetic X-ray images can be challenging as it requires high computation resources and is time-consuming. In this chapter, we first start by discussing the advances in synthetic X-ray image generation methods. Secondly, we introduce image super-resolution with deep learning, which enables us to create higher-quality synthetic X-ray images from lower-quality ones. And then, we introduce the objectives and major contributions of this work. Finally, we give the outline of this thesis.

1.1 Synthetic X-rays

The discovery of X-rays by German professor Wilhelm Röntgen [1] on November 8th, 1895, marked the rise of new emerging techniques in the medical imaging field. The technique has been used quickly in both clinics and surgical operations within months. In the past century, various applications and techniques based on X-ray technology have emerged. Projectional radiograph, also known as conventional X-ray, is a two-dimensional image produced by traditional X-ray machines [2]. It is extensively used for diagnostic purposes for different organs or structures such as chest, breast, abdomen, and head. Computed tomography (CT) was invented in 1972, and it can produce a section of the human body instead of only one view. In other words, it can provide a “slice” view from the inside of the human body, and these slices can be combined into a three-dimensional volume [2]. Fluoroscopy is another technique that can generate real-time X-ray images to see the internal structure of the human body. It is not only beneficial for diagnosis but also therapy such as interventional radiology and image-guided surgery.

Synthetic X-ray image generation or digitally reconstructed radiographs (DRRs) is an important technology in 2D/3D registration in the field of image-guided therapy (IGT) [3]. Synthetic X-ray images are simulated X-ray images projected from computed tomography (CT) data by calculating the attenuation values through Hounsfield units (HU) in CT. An arbitrary view synthetic X-ray image can be generated by rotating the CT data volume, as shown in Figure 1-1. In recent years, synthetic X-ray images have also been used in applications other than registration. For example, generating arbitrary view synthetic X-ray images from CT data are used in virtual reality (VR) simulation for training medical doctors on fluoroscopy-guided intervention procedures [4], [5] as shown in Figure 1-3. The VR training simulation intends to replace real-life training sessions, which are costly and exposed to radiation. But the performance of such simulation is often limited by the quality of synthetic X-ray images. High-quality DRRs can be used as training data for machine learning approaches in fluoroscopy-guided procedures [6]. Synthetic X-ray images can also help train deep learning models to build arbitrary view X-ray images for automatic anatomical landmarks detection, which can benefit surgical decision-making with additional 3D information [7]. Additionally, they can be used as training data for reconstructing CT data from biplanar X-rays [8]. It's also been found that using synthetic X-ray images can reduce up to half the amount of fluoroscopic images taken during actual fluoroscopy-guided intervention procedures [9]. Taking fewer fluoroscopic images is beneficial because both patient and physician can expose to less radiation.

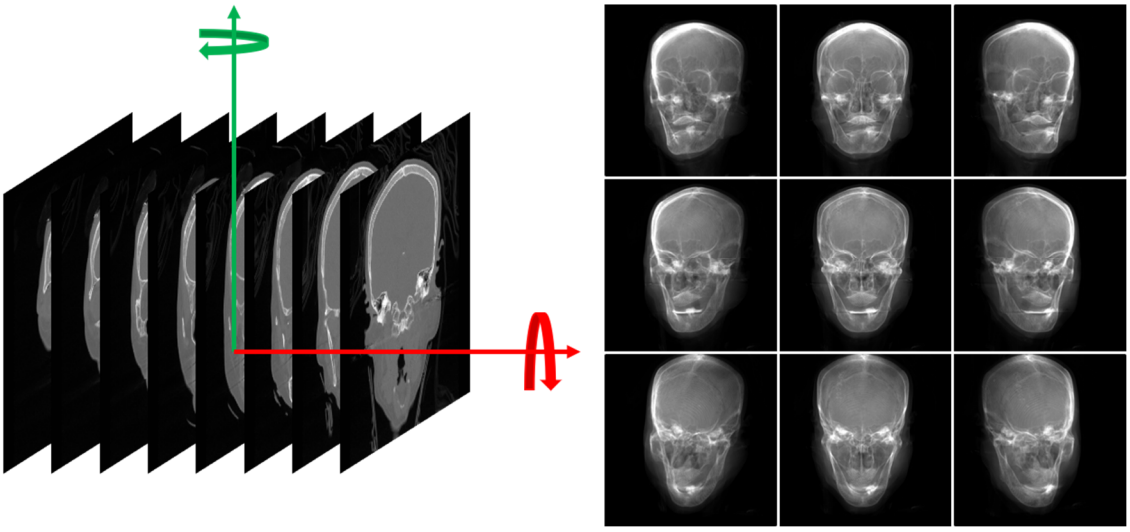

Figure 1-1: An example for arbitrary view synthetic X-ray images generation from CT data.

We require much more synthetic X-ray data with emerging technologies and the data driving deep learning approaches. Methods for synthetic X-ray image generation or DRRs can be grouped in analytic and statistical approaches, i.e., ray-tracing and Monte Carlo (MC) simulation, respectively [6]. While Monte Carlo (MC) simulation provides the most accurate simulated X-rays, it is not feasible considering the expensive computation cost. Previous works have focused on using ray-tracing approaches for efficiency considerations [10], [11]. However, they are still computationally expensive, and such ray-tracing methods often compromise image quality for speed, as shown in Figure 1-2.

In recent years, researchers also tried improving image quality by using deep learning models and more computational resources to simulate more X-ray characteristics [6] [12]. This enables the generation of highly realistic synthetic X-ray images. However, DeepDRR [6] is very complex and requires vast training data, which are not always available. Another more straightforward approach with deep learning is to use image-to-image translation to transfer the real X-ray image style to synthetic X-ray images, RealDRR [12]. It can generate more realistic synthetic X-ray images but is not highly accurate, especially when Generative Adversarial Network (GAN) is utilized. There is no guarantee that it matches the input image's anatomical structure, and artifacts can normally be found in the image. Our approach starts by making fine-quality synthetic X-ray images with lookup tables from CT data. It then uses deep learning-based super-resolution to increase the quality but focusing on the correctness of anatomical structure and fewer artifacts instead of plausible looking.

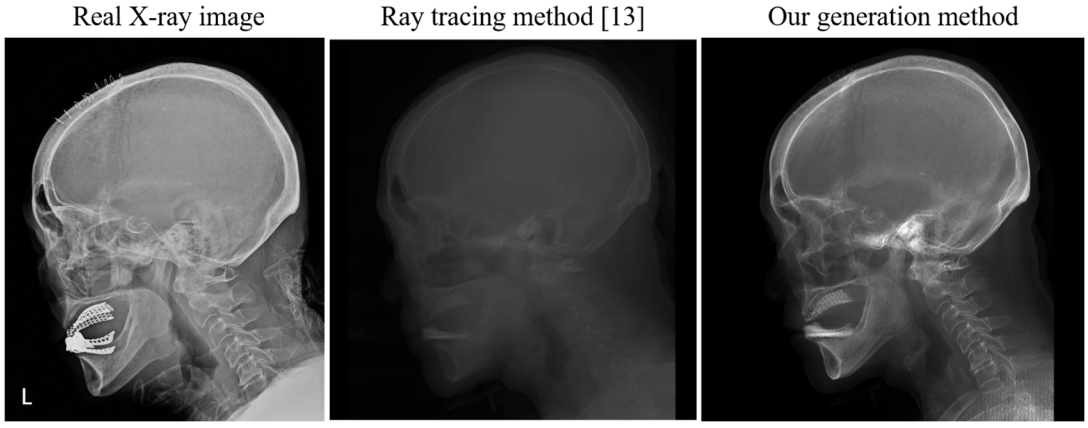

Figure 1-2: Comparison between real X-ray image, a conventional ray-tracing method from MeVisLab [13], and our generation method resulting image.

In general, there are three limitations to high-quality synthetic X-ray image generation. Firstly, it demands higher computational resources to produce high-quality synthetic X-ray images, which challenges the use in clinics or hospitals. Secondly, the algorithms for generating high-quality synthetic X-ray images are often time-consuming or complex. Finally, the resolution of the scan, i.e., the thickness of the scan, is not always dense enough due to high dose radiations of CT scanning. Each CT scan image can have high resolution, but the gap between slices can cause a lot lower resolution. An essential requirement for generating adequate synthetic X-ray images is that the resolution is approximately equal in all directions [14]. For example, if a CT scan is done in the coronal view, the projected view on the sagittal view could have less resolution. We can mitigate this issue with interpolation methods such as linear interpolation and spline interpolation. More sophisticated methods such as semantic interpolation [15] could also work well for medical image interpolation. It is also beneficial to up-sample the view with less resolution using super-resolution methods. In this thesis, we choose to use linear interpolation in our experiments and aim to develop a super-resolution method for synthetic X-ray images.

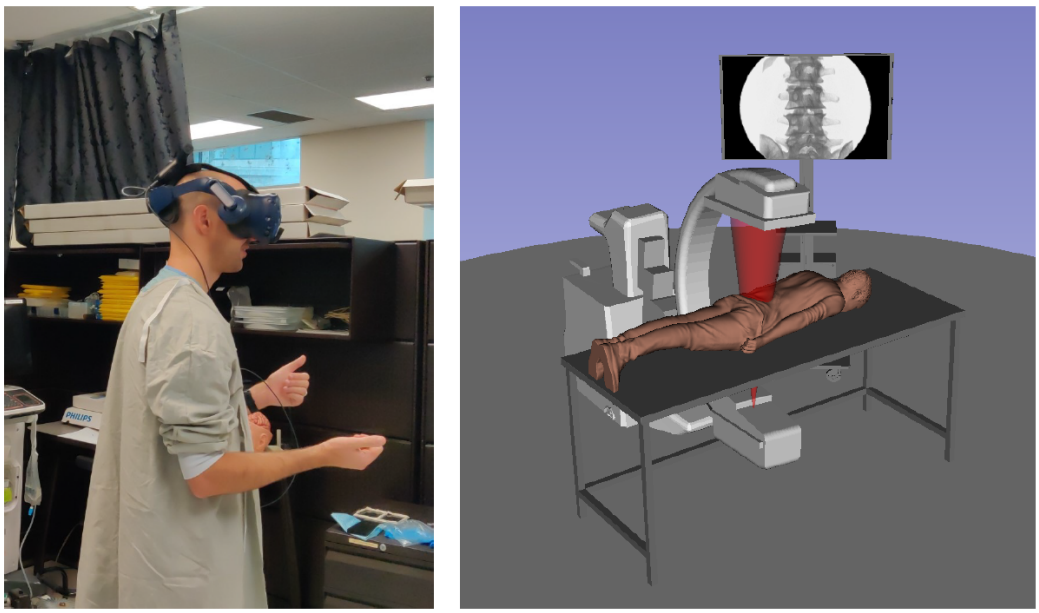

Figure 1-3: A study on VR simulation approach training for C-arm-guided intervention procedures. Left: A user wearing the VIVE Pro HMD using the VR system. Right: A screenshot of the virtual scene. The figure is reprinted from [4] with permission.

1.2 Super-Resolution

Deep learning-based super-resolution has been widely applied for natural images. It has achieved promising performance in natural image super-resolution by learning a nonlinear mapping function from low-resolution (LR) images to high-resolution (HR) images [15]. The advances in deep learning super-resolution methods have shown superior performance over conventional methods. This could also benefit research in the medical imaging field. Efforts have been made in applying super-resolution on medical images, such as CT, MRI, or X-ray [16]–[18]. High-quality synthetic X-ray images could be costly to produce and sometimes not applicable for specific applications. Super-resolution for synthetic X-ray images could bring improvements on state-of-the-art technologies such as C-Arm-based Interventions simulation with Virtual Reality (VR) by providing higher-quality synthetic X-ray images with almost no increase in processing time. It has also been found that using synthetic X-ray images can reduce up to half the amount of fluoroscopic images taken during fluoroscopy-guided intervention procedures [9]. Both approaches require near real-time processing, which makes directly generating high-quality synthetic X-ray images not applicable. An abbreviation table with explanations is given in Table 1 to understand super-resolution terms easier.

Table 1: Super-resolution abbreviation and its explanations

Abbreviation	Definition	Explanation
LR	Low-resolution image	LR images are the super-resolution network input images that are downsampled from HR images.
HR	High-resolution image	HR images are used to create LR images to form an image pair in the dataset. It is also known as GT image.
SR	Super-resolution image	SR images are the results generated by super-resolution networks.
GT	Ground Truth image	GT images are HR images used to compare the image quality with SR images.
Ref	Reference image	Ref image is used in RefSR to provide HR texture to restore the input LR image.

There are two main super-resolution method categories: Single Image Super-Resolution (SISR) and Reference-based Super-Resolution (RefSR). SISR aims to reconstruct a high resolution (HR) image from a single low resolution (LR) image [19]. RefSR works on learning the high-resolution texture details from a given reference high-resolution image [20]. In recent years, convolution neural networks (CNN) and Generative adversarial network (GAN) methods have been well explored for SISR. CNN models can reach high performance on evaluation metrics but produce overly smooth images with coarse details. On the other hand, GAN models generate appealing images with fine details but with more artifacts. This issue is especially crucial in medical imaging as image details impact diagnosis and decision-making during operations. RefSR can learn fine texture details from the reference image but still generates some artifacts and noise. Fourier transform is rarely utilized in the field of super-resolution. We propose to use a frequency domain loss as a constraint to mitigate this problem.

Deep learning-based super-resolution methods have exhibited promising performance, but there are certain limitations for existing super-resolution (SR) methods. Super-resolution datasets are usually built by downsampling HR images with a specific kernel (often bicubic kernel) to create HR and LR image training pairs. The SR model aims to learn the mapping from such image pairs. Test images are also produced from HR images using the same downsampling kernel that creates the dataset for training. The Adaptation problem happens when the testing data is not downsampled using the same downsampling kernel or degradation methods used during training. The resulting image of each degradation method can be very different, which means the model trained with one kernel does not work well on the testing data created by other kernels. This is because most SR methods only learned one kind of degradation mapping between LR and HR images. In this work, we train our network with multiple kernels to generalize the model and add the dual regression loss to improve network performance further.

1.3 Objectives

The objectives of this work focus on high-quality synthetic X-ray image generation. The objectives in detail are as follows:

- Develop and implement an efficient, high-quality and versatile synthetic X-ray image generation algorithm based on CT data

- Build synthetic X-ray image datasets for deep learning-based super-resolution methods.
- Develop deep learning super-resolution networks for producing high-quality synthetic X-ray images.
- Extend the study into application scenarios and propose possible solutions.

1.4 Contributions

The contributions of this thesis are as follows:

- Build a synthetic X-ray generation method in an alternative way that achieves good image quality and efficiency.
- Create two RefSR X-ray image datasets: a synthetic X-ray image dataset built with our generation method from real CT data and a real X-ray image dataset built from real Chest X-ray data.
- Develop a reference-based super-resolution architecture for synthetic X-ray images. It utilizes frequency domain loss to improve the resulting image quality.
- Develop an adaptation network to mitigate the performance drop when testing images are not produced using the same downsampling kernel used during training.

1.5 Thesis Outline

The rest of this thesis is organized as follows:

In Chapter 2, we illustrate the background study related to this thesis. We cover various aspects, including medical imaging, neural networks, image processing, and evaluation metrics.

In Chapter 3, we review the related works regarding the work done in this thesis. This including synthetic X-ray image generation methods, single image super-resolution, reference-based super-resolution, the usage of Fourier transform in neural networks, and research working on the adaptation issue of super-resolution.

Chapter 4 introduces our synthetic X-ray image generation method.

Chapter 5 describes our contributions on synthetic X-ray image super-resolution, including frequency-domain loss and an adaptation network.

Chapter 6 provides the setup and results of the experiments in this work, including synthetic X-ray image generation, synthetic X-ray image super-resolution, and synthetic X-ray image super-resolution adaptation network.

Chapter 7 presents one of the application scenarios for synthetic X-ray images. We applied our synthetic X-ray image generation method in Unity3D to build a VR training simulation for C-arm-guided intervention procedures.

Chapter 8 concludes the thesis with a summary of contributions and presents several possible directions for future works.

Chapter 2. Background Study

In this chapter, we introduce studies in various subjects related to this work, including medical imaging, advances in neural networks, image processing techniques, and evaluation metrics. Firstly, we introduce some concepts in the medical field, such as CT and Fluoroscopy procedures. Understanding the properties of CT images is essential for synthetic X-ray image generation.

Secondly, we explain two famous neural networks commonly used in super-resolution: Convolutional neural networks (CNNs) and Generative adversarial networks (GANs). Then, we present three commonly used loss functions in super-resolution methods: pixel loss, perceptual loss, and adversarial loss. After that, the attention mechanism for neural networks is explained. We then introduce a modern network architecture, Transformer, that is based on the attention mechanism. We also elaborate on our baseline model Texture Transformer Network for Image Super-Resolution (TTSR).

Thirdly, we talk about image processing in the spatial and frequency domain. After this, we explain the basics about Fourier transform, including discrete Fourier transform and fast Fourier transform. We also briefly introduce how they apply in image processing tasks. Then, we describe a few rescaling methods and their differences. In the end, we discuss a few evaluation metrics for super-resolution and present two of the most commonly used ones, which are used in this work to evaluate our results for synthetic X-ray image super-resolution.

2.1 Computed Tomography

Computed tomography (CT) became applicable to clinics in the early 1970s and then widely used in the diagnostic field. CT images are produced by passing X-rays through the human body at a large number of angles by rotating the X-ray tube around the body [21]. A CT scan can produce slice images of the human body with the help of the machine's computer. These slices can combine to form a 3D volume of the patient, which provides more diagnostic details. A modern CT scan can be acquired in different anatomic planes without moving the patient. The standard orientation of CT is normally the axial plane, and data can be reformatted into the sagittal or coronal plane. Images

obtained from these planes are also called different views, i.e., sagittal (SAG), coronal (COR), and axial (AX) views.

The image quality of CT scans can be determined through various characteristics: spatial resolution, contrast resolution, and temporal resolution. Spatial resolution refers to the ability to differentiate two nearby objects. It is decided by focal spot size, pixel size, slice thickness, etc. [22]. Contrast resolution refers to the ability to distinguish between differences in image intensity [23]. The contrast resolution for CT usually is not high due to the minimal differences of attenuation for tissues. Temporal resolution refers to the ability to resolve fast-moving objects and can be regarded as the shutter speed for a camera [14]. It is more important when the imaged objects are in motion, such as CT for cardiac. In our work, spatial resolution is our main concern for generating synthetic X-ray images. It is crucial when we only have one view of the CT scan. As shown in Figure 2-1, the slice thickness could affect the quality of the synthetic X-ray image projected from particular viewpoints.

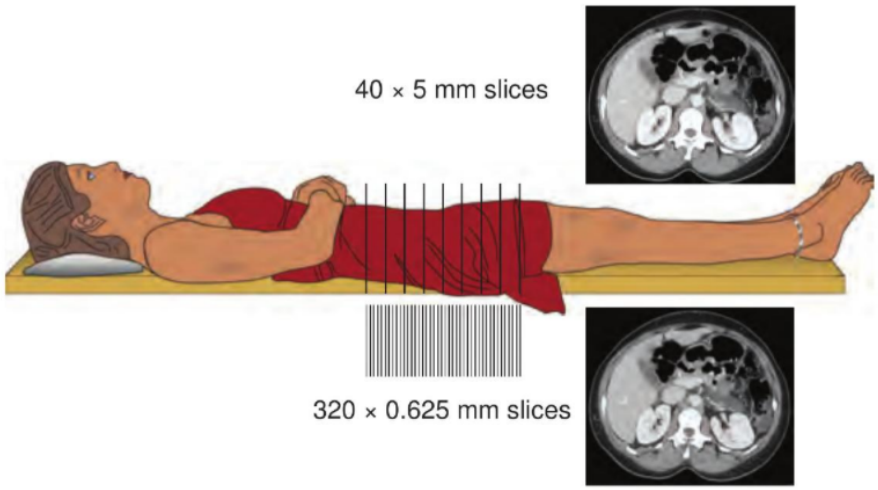

Figure 2-1: Slice thickness is an important factor in the CT image quality, which impacts the quality of the generated synthetic X-ray images. The slice thickness is related to radiation dose and CT image noise. Thinner slices yield higher spatial resolution but with noisier CT images and more radiation. Thicker slices cause synthetic X-ray image quality drops for certain angles of view. The figure is reprinted from [21] with permission.

A disadvantage of CT is that CT scans cause much more radiation than conventional radiographs since CT scans take more X-ray images. Evidence shows that the organ doses corresponding to a common CT study (two or three scans) result in an increased risk of cancer [24].

The radiation is also higher when the scan slice gap is thinner. Such concerns with radiation could be mitigated with the advances in low-dose CT technology.

2.1.1 Hounsfield Unit

Hounsfield units (HU) is a relative quantitative measurement of radio density used by radiologists to interpret CT images [25]. It is calculated from a linear transformation according to the measured CT attenuation coefficient. It is defined based on densities of air and pure water. Air and water are arbitrarily assigned to values of -1000 and 0 HU, respectively. It is computed using the following equation:

$$HU = 1000 \times \frac{\mu - \mu_{water}}{\mu_{water} - \mu_{air}} \quad (2\text{-}1)$$

where μ is the linear attenuation coefficient of voxels in CT. μ_{water} and μ_{air} are the linear attenuation of water and air. The value of HU determines the image grayscale intensity when converted into a digital image. The higher the HU, the brighter it displays in the CT image. Human bones are the whitest, and soft tissue and fat are darker in CT images. An example of HU values for head CT scans is shown in Table 2. The pixel value range is between 0 to 255 in a digital grayscale CT image. Each pixel value represents different tissues and organs attenuation, which corresponds to the HU values.

Table 2: Example of approximate HU values for tissues commonly found on head CT scans [26].

Hounsfield units	Tissue
1000	Bone, calcium, metal
60 to 100	Intracranial hemorrhage
35	Gray matter
25	White matter
20 to 40	Muscle, soft tissue
0	Water
-30 to -70	Fat
-1000	Air

2.1.2 DICOM Files

DICOM files are images saved in Digital Imaging and Communications in Medicine (DICOM) standard. It was widely used to store and exchange medical images such as radiography, ultrasonography, computed tomography (CT), and magnetic resonance imaging (MRI). A DICOM file preserves the image information and the data attributes of the image and patient. For example, it can store information such as patient information, series information, view position, slice thickness, spacing between slices, etc.

The DICOM file provides rich information for various image processing tasks, but it also has some limitations. Pixel data inside a DICOM file is compressed and requires specific packages installed to read. This creates some compatibility issues where some packages may not be able to handle specific data. In this work, we gain the needed attribute information from the DICOM file. And then, we use MicroDicom viewer to convert all DICOM files into PNG image files before any processing starts. Portable Network Graphics (PNG) is a widely used raster image file format that supports lossless data compression. A raster image is an image with a matrix structure that represents a grid of pixels.

2.2 Fluoroscopy and its Procedures

Fluoroscopy refers to the continuous acquisition of a sequence of X-ray images over time, which can be regarded as a real-time x-ray movie of the patient [21]. Fluoroscopy enables physicians to gain access to real-time patient radiographs, which is very helpful when real-time visual feedback is necessary. Thus, it can help for diagnostic and is also widely used during surgical and intervention procedures.

Fluoroscopy can be used in various tasks with different configurations. For example, some common tasks can be orthopedic surgery, podiatric surgery, etc. A type of fluoroscopy-guided intervention procedure related to this work is a C-arm-guided procedure. The configuration uses a mobile C-arm machine that can be used in surgery or pain management. In Chapter 7, we briefly introduce how our synthetic X-ray image generation method works with a VR training simulation for C-arm-based image-guided intervention procedures.

Fluoroscopy could also carry risks of high dose radiation similar to CT, especially for more complicated intervention procedures. Some recent works have been using synthetic X-ray images as a replacement to reduce the number of radiographs taken during surgical procedures [9], [27].

2.3 Convolutional Neural Networks

Convolutional neural networks (CNNs) are designed to process complex data in the form of multiple arrays, for example, a natural RGB image composed of 2D arrays corresponding to the three color channels [28]. CNNs are one of the most popular deep learning models. The modern deep learning history is occupied by famous CNNs architectures such as AlexNet [29], VGG [30], InceptionNet [31], ResNet [32] and DenseNet [33]. They are widely used in computer vision and have applications in various tasks, such as image classification, object detection, segmentation, and image reconstruction. CNNs can have much deeper architecture and learn more complex information than its predecessor Multilayer Perceptron (MLP). It is also more effective and easier to train by its characteristics such as local connection, shared weight, and not fully connected layers.

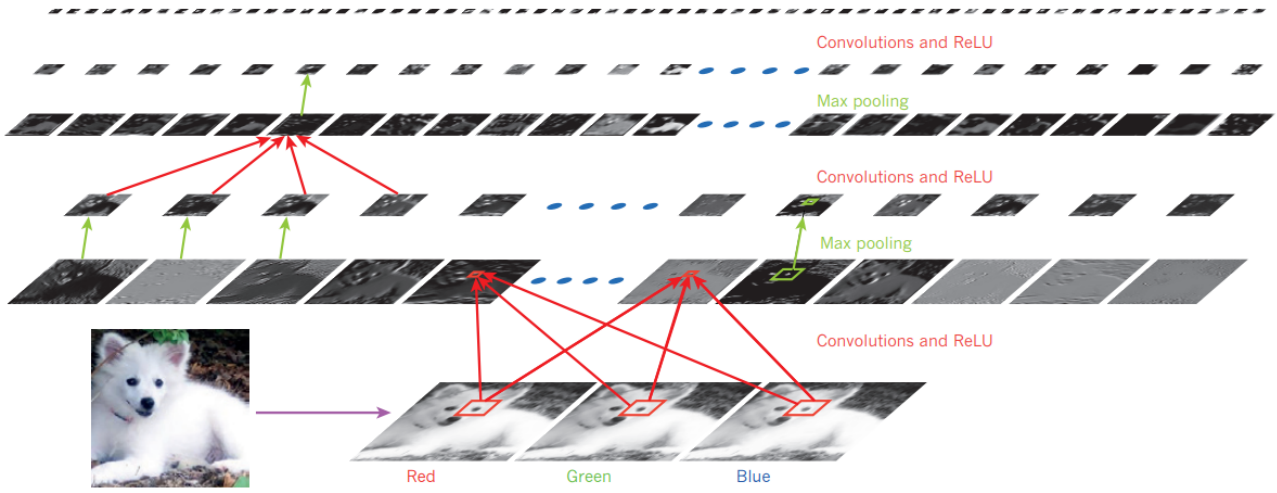

Figure 2-2: A typical CNNs architecture applied to the image of a Samoyed dog. The figure is reprinted from [28] with permission.

An example of CNNs architecture is shown in Figure 2-2. It typically starts with a few convolutional layers and pooling layers, as shown in the example. After stacking a few convolutional layers and pooling layers, more convolution and fully connected layers follow, which enable the CNNs to handle various complex tasks.

CNNs are the first type of neural network used in super-resolution in [34], where the proposed SRCNN achieved superior results than conventional SR methods. CNN-based super-resolution method remains a very successful super-resolution approach with the advances in CNNs techniques such as residual learning and attention mechanism.

2.4 Generative Adversarial Networks (GAN)

Generative adversarial networks (GANs) [35] distinguish from CNNs as it contains two neural networks: generator and discriminator. It was proposed by Goodfellow et al. [35] in 2014 and started a new era in deep learning research. The idea of two neural networks competing with each other enables GAN to generate realistic and indistinguishable data compared to training data. The generator always tries to generate more realistic data, while the discriminator learns to differentiate the generated data and training data.

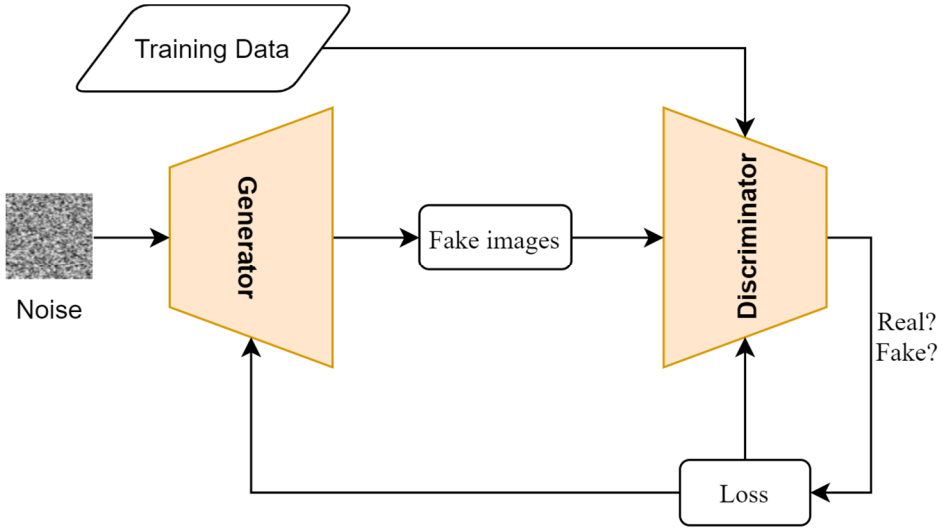

Figure 2-3: A typical GAN architecture

A typical GAN architecture is shown in Figure 2-3. The generator's input is a random noise vector, and it is mapped to a multi-dimensional vector by the generator. The discriminator serves as a classifier that takes the real image training data and fake images generated by the generator as input. The loss from it is then fed to both generator and discriminator to facilitate the training process. A variety of GAN variants emerged and greatly improved the training process's stability and GAN's ability to generate specific categories of data or higher quality data. Some representatives of famous

GAN variants including cGAN [36], DCGAN [37], CycleGAN [38], WGAN [39] , BigGAN [40], StyleGAN [41] , etc.

GAN-based deep learning super-resolution method also gained a tremendous amount of research interest over the past few years. It shows great potential for improving the perceptual quality of the predicted HR image with a scarify on generating some artifacts.

2.5 Loss Functions

A loss function is used to evaluate the prediction error of the neural network. We also use it to optimize the neural network. We want to minimize the loss during training to gain better performance. Loss functions are used to measure the reconstruction error and optimize the model performance in the super-resolution field [42]. In this section, we introduce a few loss functions used in super-resolution methods.

2.5.1 Pixel Loss

Pixel-wise loss is the most straightforward loss function used to understand the differences between images at the pixel level. L1 and L2 loss are two widely used pixel loss functions in image reconstruction tasks such as super-resolution. L2 loss was more widely used in early times in super-resolution, but researchers later found out that L1 loss yields sharper results and is easier to converge [19], [20], [43]. L1 loss calculates the Mean Absolute Error (MAE) between two images. Given N as the number of pixels in an image, I^{HR} and I^{SR} are the HR image and up-sampled SR image, it can be defined as:

$$\mathcal{L}_1 = \frac{1}{N}\sum_{i=1}^{N}|I^{HR} - I^{SR}| \tag{2-2}$$

L2 loss calculates the Mean Squared Error (MSE) between two images, can be defined as:

$$\mathcal{L}_2 = \frac{1}{N}\sum_{i=1}^{N}\left(I_i^{HR} - I_i^{SR}\right)^2 \tag{2-3}$$

2.5.2 Perceptual Loss

Perceptual loss [44], also known as content loss, measures higher-level differences rather than pixel loss. It can evaluate the perceptual quality of images by measuring the semantic differences

between images using a pre-trained image classification network [42]. The pre-trained network is usually a VGG [30] or ResNet [32]. Perceptual loss can be interpreted as the Euclidean distance between high-level representations. Given the network as ϕ, the extracted representations on i-th layer as ϕ_i, the perceptual loss can be defined as:

$$\mathcal{L}_{per} = \frac{1}{C_i H_i W_i} \|\phi_i(I^{HR}) - \phi_i(I^{SR})\|_2^2 \tag{2-4}$$

where (C_i, H_i, W_i) denotes the channel numbers, height, and width of the representation on layer i. Super-resolution methods that adopt perceptual loss can produce more visually appealing results. It is widely used in GAN-based super-resolution approaches to achieve better perceptual quality.

2.5.3 Adversarial Loss

Adversarial loss [45] refers to the loss used in a GAN model that consists of a generator and discriminator. GAN's training process can be understood as an iteration among two stages: a stage that fixes the generator and trains the discriminator to make better predictions between real or fake; and a stage that fixes the discriminator and trains the generator to fool the discriminator. Such adversarial training can produce realistic data such that, in the end, the discriminator can not distinguish between real and fake data.

Super-resolution can benefit from adversarial training and adopts it by assuming the SR generation model is a generator where the input is an LR image instead of random noise. Here we use SR image refer to the resulting image of super-resolution methods and ground truth (GT) image for the high-resolution image target we want to restore. Thus, we can use a discriminator to distinguish between the generated SR image and the GT image. For example, Enhancenet [46] adopts adversarial loss based on cross-entropy as follows:

$$\mathcal{L}_G = -\log D(G(I_{LR})) \tag{2-5}$$

$$\mathcal{L}_D = -\log D(I_{HR}) - \log (1 - D(G(I_{LR}))) \tag{2-6}$$

where $\mathcal{L}_G$ and $\mathcal{L}_D$ are the adversarial loss of the generator and discriminator. I_{LR} is the generator input image. I_{HR} is the high-resolution GT image.

2.6 Attention Mechanism

Attention mechanism [47] was first proposed in Natural Language Processing (NLP) field for machine translation. It was used to solve the Seq2Seq [48] model issue that the performance drops when handling long sequences, and giving each word the same weight in a sentence is inefficient. The attention mechanism allows the model to access the entire sequence and focus on more relevant information. With the effectiveness in the NLP field, the attention mechanism was quickly introduced into the computer vision field. We show an example of visual attention in Figure 2-4.

Figure 2-4: An example using attention mechanism in caption generation task. A model with vision attention can learn the most related object in an image related to a word. Images used are from [49].

The vision attention mechanism can be regarded as simulating the human perception process. When the human visual system takes in visual information, we automatically focus on more important regions. For example, we can distinguish vital objects from the background and adjust the focal point for close and far objects. The attention mechanism is proven to improve network performance for multiple tasks efficiently. This is followed by exploration for a variety of attention mechanisms such as soft and hard attention [49], self-attention [50], global and local attention [51]. These attention mechanisms are also applied to several networks, including recurrent neural networks (RNNs), CNNs, GAN, and Transformer.

2.7 Transformer

Transformer [50] is a network architecture proposed by Vaswani et al. in 2017, which is solely based on attention mechanisms. The architecture does not use any convolution or RNNs, which was the mainstream in Natural Language Processing (NLP). Transformer does not require the input to be

processed in order, which enables a new level of parallelization that can train much faster than its predecessor. It quickly becomes the mainstream for NLP tasks and promotes modern pretrained NLP systems such as BERT [52] and GPT [53].

Transformer has been a new direction to explore in computer vision, as it achieved considerable success in NLP fields. Carion et al. [54] proposed Detection Transformer (DETR), an object detection system based on transformer which achieved good performance on large objects. Vision Transformer (ViT) [55] adopted transformer on image recognition tasks where it processed images as a sequence of 2D patches. Chen et al. [56] proposed image processing transformer (IPT) for image reconstruction tasks, including denoising, super-resolution, and deraining. TTSR [20] adopted a transformer architecture for reference-based image super-resolution and achieved superior results compared to previous works.

2.7.1 TTSR

In this thesis, we use Texture Transformer Network for Image Super-Resolution [21] (TTSR) as our baseline RefSR model. It utilized high-resolution reference (Ref) images to restore LR images and boost the results' quality. A Ref image could be selected from adjacent frames in a video or images from different viewpoints [57]. This paper is one of the first to introduce transformer architecture into image generation tasks. The proposed texture transformer utilizes the advantage of transformer to exchange high-resolution texture from Ref Images to LR images. Cross-scale feature integration is also proposed to improve model performance, where multiple texture transformers of different scales can be stacked together to exchange information. We introduce more details in the following sections.

2.7.1.1 Texture Transformer

The structure of the proposed texture transformer is shown in Figure 2-5. The texture transformer uses four images as input: LR, Ref, $Ref\downarrow\uparrow$ and $LR\uparrow$. These four images are different variants of LR and Ref images. $LR\uparrow$ is the upsampled version of the LR image. $Ref\downarrow\uparrow$ is produced by first downsampling the Ref image and then upsampling it again. Such rescaling operation aims to match the useful texture location in the Ref image for texture exchange with the LR image. The LR image is also processed by a DNN backbone to extract the features. Four innovative methods are

proposed in texture transformer to make transformer suitable for reference-based super-resolution tasks: learnable texture extractor, relevance embedding, hard attention, and soft attention.

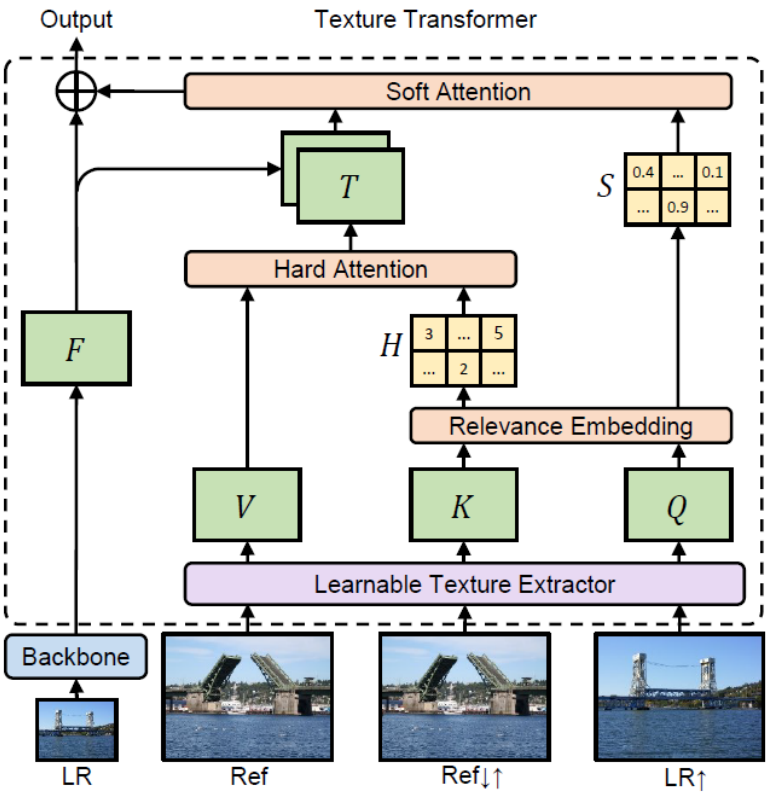

Figure 2-5: The proposed texture transformer in [20]. Q, K, and V are the texture features extracted from an up-sampled LR image, a sequentially down/up-sampled Ref image, and an HR Ref image, respectively. H and S refer to the hard/soft attention map, calculated from relevance embedding. F is the LR features extracted from a DNN backbone and is further fused with the transferred texture features T for generating the SR output. The figure is reprinted from [20] with permission © 2021 IEEE.

- Learnable texture extractor

Learnable texture extractor (LTE) enables a joint feature learning between LR and Ref image, and such feature embedding also creates the foundation for applying the attention mechanism. Texture extraction is important for a more accurate texture information exchange which helps generate SR images. Previous works [57] used semantic features extracted by a pre-trained classification model such as VGG [30]. Instead, the parameters of the LTE are updated during the training process, which improves the accuracy when extracting features. The LTE outputs, Q (query), K (key) , and V (value) correspond to the three elements of attention mechanism, which are further used in the other three methods in texture transformer.

- Relevance embedding

Relevance embedding works on estimating the relevance between the LR and Ref image. This is achieved by comparing the similarity between Q and K. It formulates the extracted features from the LR and Ref image as the Q and K in the transformer to obtain a hard-attention map and a soft-attention map [20].

- Hard and Soft attention

The hard attention and soft-attention modules transfer and synthesize HR features from the Ref image into LR features extracted from the backbone through the attention maps[20].

Hard attention transfers the HR texture features V in the reference image to the LR image. Compared to regular attention that takes a weighted sum of V for each query, the hard attention model used here only takes the most relevant value in V. This approach mitigates the blur effect commonly caused by regular attention mechanisms.

In reference-based super-resolution, effective texture transfer needs to transfer textures and emphasize more relevant textures. Soft attention is used to synthesis features from the output of hard attention T and the LR images feature F. During this process, more relevant textures are enhanced, and less relevant ones are relived.

2.7.1.2 Loss Functions used in TTSR

TTSR adopts three loss functions in their approach. The general concept of these loss functions is introduced in Section 2.5. Given λ as the loss weight coefficient, the overall loss of TTSR is as follows:

$$\mathcal{L}_{overall} = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{per}\mathcal{L}_{per} \tag{2-7}$$

The reconstruction loss utilizes a $L1$ loss. Given C, H, W as the channel, height, and width of an image. The reconstruction loss between HR image I^{HR} and LR image I^{LR} can be defined as:

$$\mathcal{L}_{rec} = \frac{1}{CHW}\|I^{HR} - I^{SR}\|_1 \tag{2-8}$$

TTSR adopts the adversarial loss in WGAN-GP [58], which yields more training stability and better performance. The loss function is defined as:

$$\mathcal{L}_D = \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] - \mathbb{E}_{\tilde{x}\sim\mathbb{P}_r}[D(x)] + \lambda\mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(\|\nabla_{\hat{x}}D(\hat{x})\|_2 - 1)^2] \tag{2-9}$$

$$\mathcal{L}_G = -\mathbb{E}_{\tilde{x}\sim \mathbb{P}_g}[D(\tilde{x})] \tag{2-10}$$

Perceptual loss is widely used in GAN based super-resolution method to improve perceptual quality. The key idea of perceptual loss is to enhance the similarity in feature space between the prediction image and the target image [20]. The perceptual loss in TTSR consists of two parts:

$$\mathcal{L}_{per} = \mathcal{L}_{fpl} + \mathcal{L}_{tpl} \tag{2-11}$$

$$\mathcal{L}_{fpl} = \frac{1}{C_i H_i W_i} \left\| \phi_i^{vgg}(I^{SR}) - \phi_i^{vgg}(I^{HR}) \right\|_2^2 \tag{2-12}$$

$$\mathcal{L}_{tpl} = \frac{1}{C_j H_j W_j} \left\| \phi_j^{lte}(I^{SR}) - T \right\|_2^2 \tag{2-13}$$

where $\mathcal{L}_{fpl}$ is the traditional perceptual loss that focuses on the feature map level. $\phi_i^{vgg}(\cdot)$ denotes the feature map of the i-th layer in VGG19. The dimension of the feature map at the i-th layer is (C_i, H_i, W_i). $\mathcal{L}_{tpl}$ is the transferal perceptual loss proposed in TTSR [20]. $\phi_j^{lte}(\cdot)$ denotes the texture feature map of the j-th layer of LTE. The shape of this layer is represented by (C_j, H_j, W_j). T is the transferred HR texture features as shown in Figure 2-5. The transferal perceptual loss enables the network to learn similar texture features and makes the model more effective in transferring the HR textures from the Ref image.

2.8 Spatial and Frequency Domain for Imaging

Images are commonly represented in the spatial domain that makes it easy for humans to perceive and understand. In image processing, the frequency domain is more convenient to detect certain patterns and features in an image. We can also obtain new information in the frequency domain. We introduce the spatial and frequency domain in the following sections.

2.8.1 Spatial Domain

Spatial domain represents images in the form of matrix. A natural RGB image can be represented with a 3D vector of 2D matrixes. Similarly, a single 2D matrix can represent a grayscale image. The spatial location of each pixel stores in the indices. The intensity of each pixel is saved in the value of the matrix. An example of a grayscale image in the spatial domain is shown in Figure 2-6.

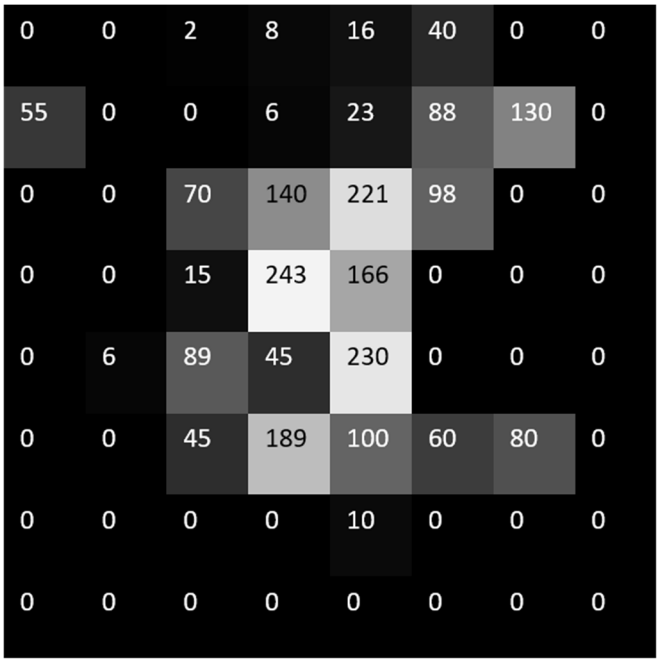

Figure 2-6: Example of an 8 x 8 grayscale image in the spatial domain

2.8.2 Frequency Domain

Frequency domain represents the changing rates of pixel values in the spatial domain. An image can be transferred into frequency domain in various methods, and Fourier transform is one of the most commonly used ones. An example of an image transferred into frequency domain using Fourier transform is shown in Figure 2-7. The frequency domain obtained by Fourier transform can also be inverted and get the same spatial domain image. The components of frequency domain can be divided into two parts: high-frequency and low-frequency components. High-frequency components are often related to sharp details and edges. Low-frequency components are more correspond to smooth areas.

In a typical image processing task, we process the image with Fourier transform and apply a filter on the frequency domain results. Then, we can obtain the processed image by computing the inverse Fourier transform. For example, we can blur an image by reducing its high-frequency components or sharpen an image by increasing the magnitude of its high-frequency components [59].

Figure 2-7: Image transferred into frequency domain. Left: spatial domain. Right: frequency domain.

2.9 Fourier Transform

Fourier transform [60] is a mathematical transform widely used in many fields of science and technology. In theory, it decomposes functions based on space or time that are decided by spatial or temporal frequency. The Fourier transform of $f(x)$ is defined as:

$$\int_{-\infty}^{\infty} f(x) e^{-i2\pi xs} dx \tag{2-14}$$

The Fourier transform is an important technique in image processing. More specifically, discrete Fourier transform is used for image processing. It is one of the methods we can use to transfer images from the spatial domain to the frequency domain. The results in the frequency domain contain rich information about frequency details, including low and high-frequency patterns.

2.9.1 Discrete Fourier Transform

Discrete Fourier transform (DFT) can be regarded as a special case of continuous Fourier transform theory. The values of the transform are available only at discrete intervals when the transform is evaluated by numerical computing [60]. And many applications involving the continuous Fourier transform rely on a digital computer for implementation, which leads to the use of discrete Fourier transform [61].

Assume we have a sequence of samples X_n, it can be processed by DFT as follows:

$$X_n = \sum_{n=0}^{N-1} x_n e^{-\frac{i2\pi}{N}kn} \quad (2\text{-}15)$$

$$\{X_n\} = x_0, x_1, \cdots, x_{N-1} \quad (2\text{-}16)$$

2.9.2 Fast Fourier Transform

The fast Fourier transform (FFT) [62] is a method for efficiently computing the discrete Fourier transform (DFT) of a sequence (discrete data samples). It takes advantage of the fact that the calculation of the coefficients of the DFT can be carried out iteratively, which results in a considerable savings of computation time [62]. FFT significantly improved the speed of Fourier transform, which makes it applicable to image processing.

FFT is commonly used to transform images between spatial domain and frequency domain. Most image processing techniques are performed in the spatial domain, but the frequency domain is also widely used in image analysis, compression, and reconstruction. Magnitude and phase represent an image processed by FFT. The magnitude (or magnitude spectrum) shows the general information about how frequency components are distributed in an image. The phase preserves the location information of the frequency components. We usually only display the magnitude in image processing, as it shows all high-frequency and low-frequency details. But we need both magnitude and phase to inverse images from frequency domain back to spatial domain.

Examples of HR and LR images in spatial and frequency domain are shown in Figure 2-8. We use the bicubic kernel to downsample the LR images from HR images with a factor of four. The magnitude spectrum shows lower frequency components in the center and higher frequency components in the surrounding area of the image. We can observe that some high-frequency details are lost in the magnitude of LR images, while the low-frequency details remain abundant.

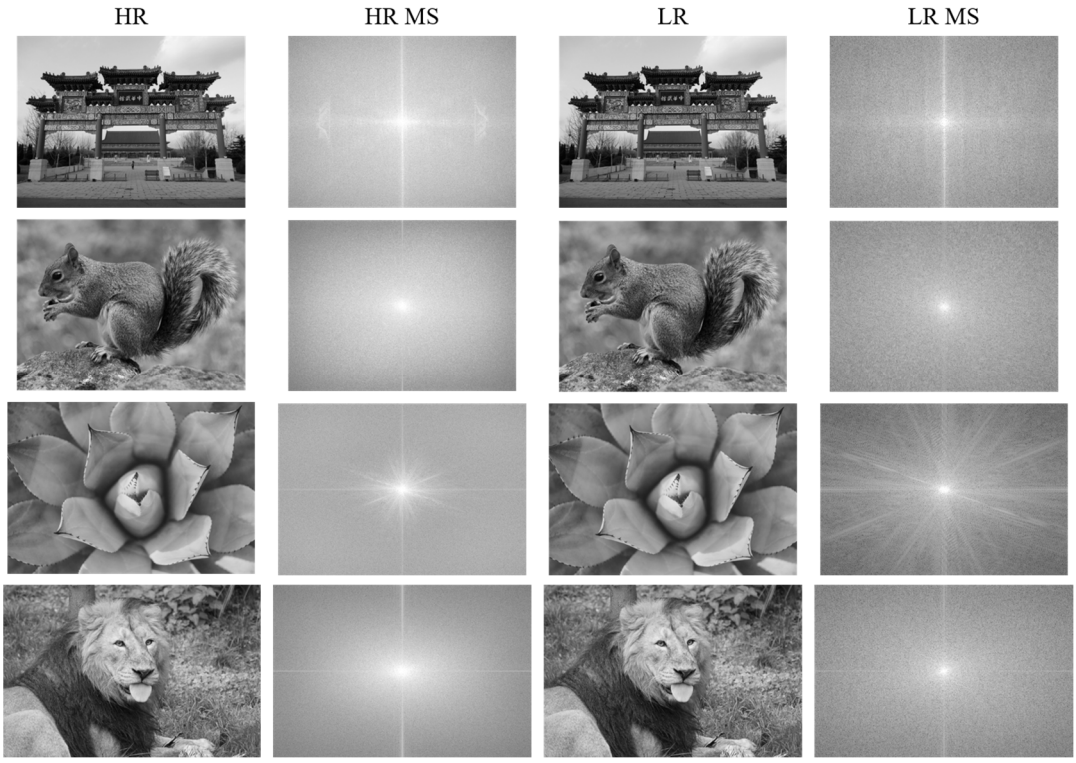

Figure 2-8: A few example images from DIV2K [63] dataset. HR image has 2K resolution. LR images are downsampled x4 from HR images.

2.10 Rescaling Methods

Rescaling or resampling is a technique used to change an image into a different resolution. Upsampling and downsampling refer to the increase and decrease of the resolution of an image, respectively. We often use downsampling to create the low-resolution images from high-resolution images to form a super-resolution dataset. Super-resolution can be regarded as a powerful learnable upsampling method. Different resampling methods have trade-offs between lossless and speed. The pooling layers in neural networks can also be regarded as downsampling methods. We introduce a few commonly used rescaling methods and pooling methods.

- Nearest Neighbor Resampling: This is one of the simplest methods for resampling. It chooses to replace the pixel value with the nearest pixel value in the original image. It is considered the most efficient method for speed but leads to some coarse and jaggedness of the resulting image.

- Bilinear Resampling: Bilinear resampling uses linear interpolation to compute the values of pixels. When upsampling or downsampling, a 2 by 2 area is considered for the new pixel value. It produces smoother results compare to the nearest neighbour method.
- Bicubic Resampling: Bicubic resampling choose 2 more pixels next to each pixel into consideration compared to Bilinear resampling. A 4 by 4 area with a total of 16 pixels is considered using a cubic spline. This is the most used method for its good trade-off between speed and accuracy.
- Pooling is used in neural networks to reduce the size of feature maps for computation efficiency. A pooling operation can perform in various sizes. For example, a 2 x 2 patch size pooling operation can reduce the feature maps by a factor of two. Some commonly used pooling methods, such as average pooling, max pooling, and min pooling, are defined by how they calculate the value for each patch of the feature map. For example, a max pooling operation keeps the maximum value of all pixels in a patch.

Our work mainly focuses on the downsampling methods because they are discussed in Sections 5.3 and 6.4 for the adaptation network. Image obtained through different resampling methods can be different in aspects such as sharpness and smoothness. Resampling methods in different software packages can also yield differences in their results. For example, common software or libraries such as MATLAB, OpenCV, and PIL (Python Imaging Library) [64] can all yield very different results for their resampling results. Those differences are mainly due to the methods used for processing the image's pixel information and the rescaling method's implementation details. The differences are sometimes hard to perceive in the spatial domain. However, we can observe the differences in the frequency domain with comparison in absolute pixel value differences, as shown in Figure 2-9. Brighter areas mean more differences to the PIL Bicubic image when comparing absolute pixel value differences. Compared to PIL Bicubic, we can observe more differences in high-frequency patterns for various downsampling methods.

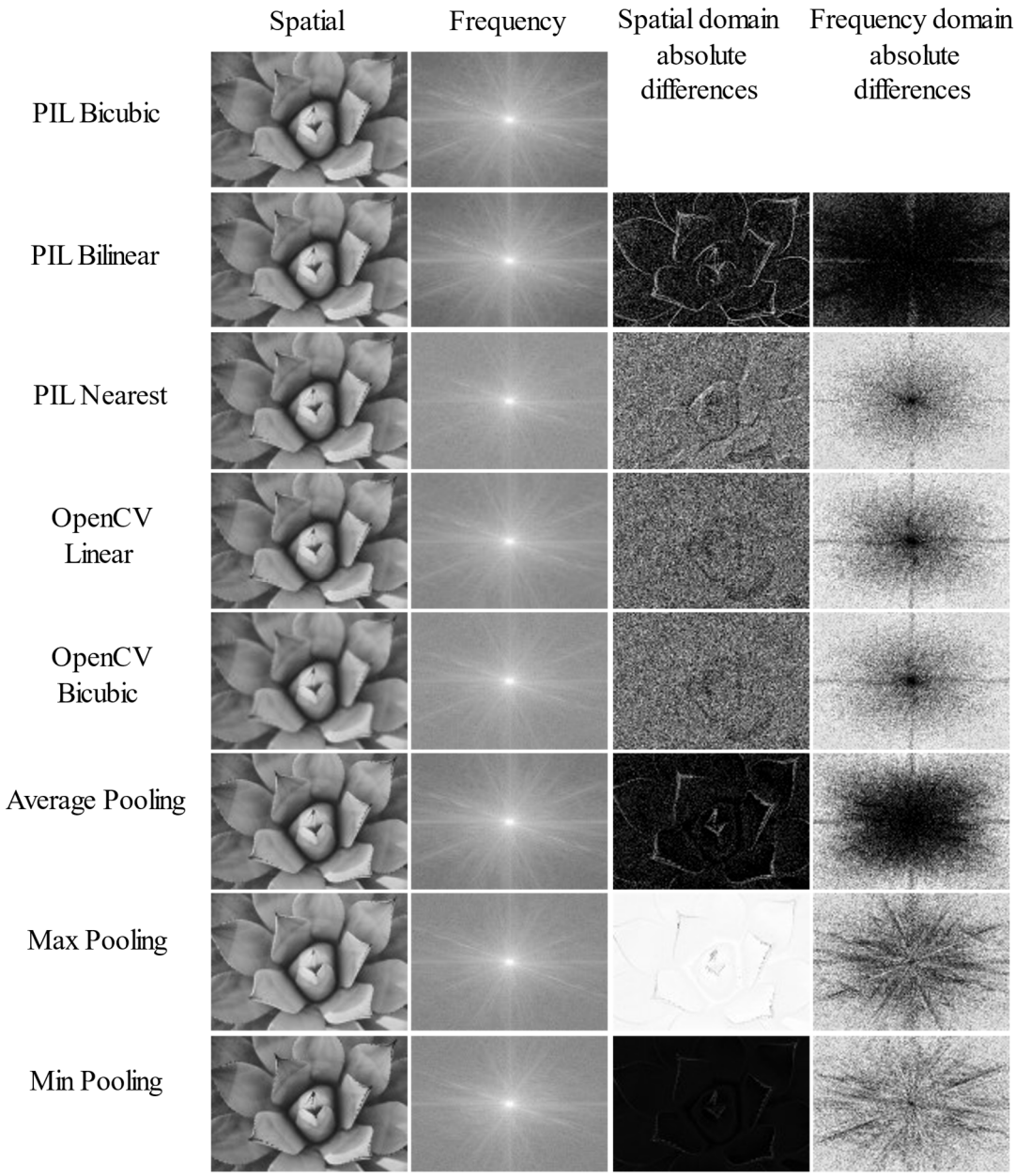

Figure 2-9: Compare different kernels results in spatial and frequency domain with absolute pixel value differences. The image is downsampled x4 using all kernels and then compared absolute differences with PIL Bicubic. The brighter the absolute differences figure is, the more different the images are compared to the PIL Bicubic image.

2.11 Evaluation Metrics for Super-Resolution

Image quality is the main objective for super-resolution methods. Image quality assessment (IQA) methods generally include subjective and objective methods. The subjective methods aim to

evaluate perceptual quality with the help of human evaluation, such as mean opinion score (MOS) and learning-based perceptual quality. The objective methods use computational measurements for evaluation, such as peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), multi-scale structural similarity (MS-SSIM) [65]. Subjective methods are ideal for super-resolution tasks but expensive and time-consuming. Thus objective methods are still the mainstream for super-resolution evaluation. PSNR and SSIM are the most widely used evaluation metrics for super-resolution. We use them in this study and introduce them in the following sections.

2.11.1 PSNR

Peak signal-to-noise ratio (PSNR) calculates the pixel-wise differences between two images. PSNR is measured in decibels (dB), and a higher PSNR value represents higher image reconstruction quality. Given L as the max pixel value of an image, the PSNR is defined as:

$$PSNR = 10 \cdot log_{10}\left(\frac{L^2}{MSE}\right) \tag{2-17}$$

where the mean squared error (MSE) is defined as:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(I_i - \hat{I}_i\right)^2 \tag{2-18}$$

where I_i denotes the ground truth image and $\hat{I}_i$ represents the super-resolution result image. N is the number of pixels in the image.

2.11.2 SSIM

Structural Similarity Index (SSIM) [66] measures the image quality from the image formation perspectives. It also measures pixel-level information but is designed to evaluate images' brightness, contrast, and structure.

$$SSIM(x, y) = \frac{2(\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)} \tag{2-19}$$

$$C_1 = (K_1 L)^2 \tag{2-20}$$

$$C_2 = (K_2 L)^2 \tag{2-21}$$

$$K_1, K_2 \ll 1 \tag{2-22}$$

where x and y are the compared two images. μ denotes the average of images, σ represents the variance of images. σ_{xy} denotes the covariance of x and y. C_1 and C_2 are two constants included to stabilize the division with weak denominators. L is the dynamic range of the pixel value (255 for grayscale image).

Chapter 3. Related Works

In this chapter, we review the literature on synthetic X-ray image generation and super-resolution methods. In the first section of this chapter, we introduce a few conventional generation methods which are primarily used for image registration. Then, we discuss some recent works that adopt deep learning to synthetic X-ray generation methods. These methods are typically used to generate high-quality training data for deep learning methods but are complex and have much higher computational costs.

In the second section, we introduce recent works in the field of super-resolution. Firstly, we cover recent advances in Single Image Super-Resolution (SISR). SISR represents the initial efforts made to apply deep learning in the super-resolution field. We discussed both CNN and GAN-based super-resolution methods, where the latter focuses more on the perceptual quality of the results. We then introduce the process that Reference-based Super-Resolution (RefSR) evolved from traditional to deep learning super-resolution methods. Thirdly, we talk about how Fourier transform got adopted in neural networks for various tasks. Finally, we discuss some approaches used to mitigate the adaptation issue that most super-resolution methods have.

3.1 Synthetic X-ray Image Generation

3.1.1 Ray Tracing-based Method

Ray tracing is a versatile technique that can generate an image by tracking the light source through pixels of objects. It is widely used in computer graphics to rendering realistic light and object interactions such as reflection and shadows. Ray tracing can also be used to simulate the X-ray projection process through the human body. A dedicated ray tracing algorithm can calculate the ray's path travelling through the human body and simulate the amount of X-ray absorbed by different tissues in the human body.

Efforts were made to convert CT volume or slicing images into synthetic X-ray images, also known as Digital Reconstructed Radiographs (DRRs), with fast rendering algorithms such as ray tracing or ray casting methods. One of the quickest projection methods was proposed by Robert L.

Siddon [10] dated back to 1985. It calculates the exact radiological path by the intersections with orthogonal sets of parallel planes instead of individual voxels of the CT array. Jacobs et al. further improved the efficiency of Siddon's algorithm by reducing the calculation for indices and ray summing process[67]. Zhao et al. [11] proposed a new ray-tracing technique that utilized the spatially recursive properties of ray tracing. However, even with the algorithm's speed improving, it is still limited to low-resolution image reconstruction. Researchers also aimed to improve the DRRs quality, specifically for CT data. Sacrificing the speed for quality, Sherouse et al. [68] adopted trilinear interpolation to consider every intersected voxel and a heuristics approach to simulate some physics of radiography. Previous works can only calculate DRRs for the energy when the CT data is acquired. Milickovic et al. [69] aimed to develop a DRR method for different photon energies and adopted volumetric data set (VDS) to improve processing speed.

3.1.2 Deep Learning-based Methods

The above methods focus more on efficiency but are not realistic compared to real X-ray images. In recent years, a few works have aimed to generate more realistic synthetic X-ray images to provide training data for deep learning methods or guiding images for VR training simulations. DeepDRR [6] proposed a framework for fast and realistic synthetic X-ray image generation. It considered the drawbacks that ray tracing could not model beam hardening and scattering. Material segmentation and neural network-based scatter estimation were used to model more X-ray characteristics. DeepDRR can produce realistic synthetic X-ray images that the trained neural network model can use directly for clinical data without re-training or domain adaptation. However, it needs a large amount of data to train several neural networks to make the precise generation. DeepDRR is suitable for producing deep learning training data but not good for VR training simulators because of the complexity and time cost. RealDRR [12] combines fast ray tracing and deep learning-based image-to-image translation. A conditional Generative Adversarial Network (cGAN) is used to translate the ray tracing result to a more realistic synthetic X-ray image. The style of the synthetic X-ray image is well-matched with real X-ray images. However, it also generates visible artifacts as a common problem for GAN-based image generation methods.

3.2 Super-Resolution

3.2.1 Single Image Super-Resolution

Single Image Super-Resolution (SISR) learns an end-to-end image mapping function represented by a CNN between low-resolution (LR) and high-resolution (HR)images[20]. Dong et al. [34] proposed Super-Resolution Convolutional Neural Network (SRCNN) that first uses the deep learning method in SISR. It achieved superior results compared to previous conventional super-resolution methods. Shi et al. [70] proposed the subpixel convolution layer to replace the deconvolution used in [34], which significantly boosts speed and performance. Another breakthrough made by Lim et al. [19] proposed enhanced deep super-resolution network (EDSR) introduced Residual Block and removed batch normalization. Zhang et al. proposed Residual Dense Network (RDN) [71] that adopted residual dense block [33] to utilize both local and global features better. Zhang et al. [72] added the attention mechanism and proposed Residual Channel Attention Networks (RCAN) to improve network performance further. Dai et al. [73] introduced second-order channel attention that enables more powerful feature correlation learning. Dual Regression Networks (DRN) [15] demonstrated that using a dual regression to form a closed-loop can increase the network performance by reducing the potential mapping space.

The above methods used CNN yield strong PSNR performance. However, they do not have excellent visual quality for human perception. GAN gradually becomes popular in the field of super-resolution due to its resulting fine perceptual quality. Johnson et al. [44] introduced perceptual loss into SISR. Ledig et al. [45] proposed a generative adversarial network for image super-resolution (SRGAN) that first adopted GAN method and showed appealing visual quality. EnhanceNet [46] combined a novel texture transfer loss and perceptual loss to enhance the texture quality in the predicted SR image. Wang et al. [74] adopted Residual-in-Residual Dense Block, and relativistic GAN further improved the perceptual quality of the results. Super-Resolution Generative Adversarial Networks with Ranker (RankSRGAN) [75] proposed to learn a ranker for perceptual metrics and used a novel rank-content loss to optimize the perceptual quality.

3.2.2 Reference-based Super-Resolution

Reference-based Super-Resolution (RefSR) takes advantage of learning more accurate details from the HR reference image. The reference (Ref) image could be selected from adjacent frames in a video, images from web retrieval, an external database (dictionary), or images from different viewpoints [57]. An example of some pairs of input and reference images is shown in Figure 3-1. The idea of reference-based or example-based super-resolution has already emerged even before the deep learning SR methods succeed. Freeman et al. [76] proposed to learn from example patches using a simplified Markov network which considered both local patch information and spatial neighbourhood effect. Chang et al. [77] proposed a neighbour embedding method inspired by manifold learning where multiple training examples can contribute simultaneously to the generation method. Freedman and Fattal [78] extended the previous example-based super-resolution framework and introduced local self-similarity that improved algorithm efficiency. Sun et al. proposed [79] scene matching to learn textures from an extensive database that contains more than six million images.

As deep learning gained success in SISR, there are also advances in RefSR methods. Zheng et al. [80] proposed a hybrid imaging system that combined an example-based approach and deep learning SISR approach [81]. It decomposed images into low frequency and high-frequency sub-bands, and then applied patch matching for high-frequency reconstruction. CrossNet [82] adopted a flow-based cross-scale warping to transfer features. Super-Resolution by Neural Texture Transfer (SRNTT) [57] adopted patch matching with VGG extracted features to swap similar textures between LR and Ref images. It managed to transfer more meaningful textures by matching textures in the feature space. Yang et al. [20] proposed Texture Transformer Network for Image Super-Resolution (TTSR), which applied a transformer [50] with soft and hard attention models that achieved superior performance over traditional SISR methods.

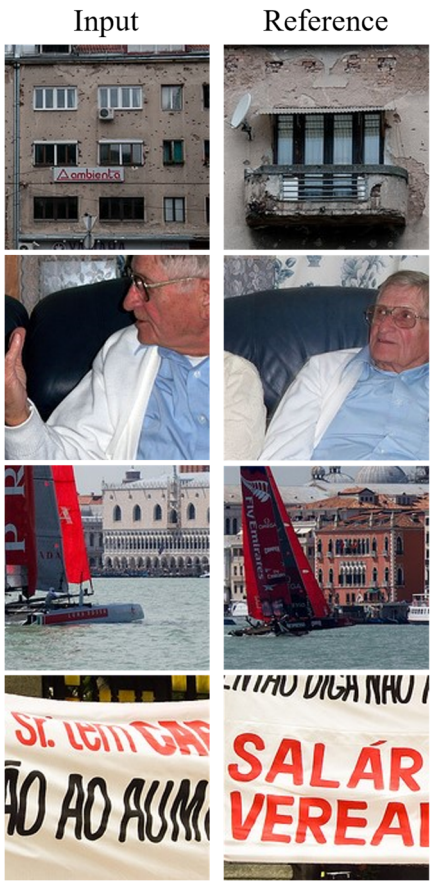

Figure 3-1: A few input and reference images examples from CUFED [57] Training dataset. The images are 160 x 160 resolution cropped from original images. The input image is downsampled as LR input during training, while the reference image provides high-quality texture to restore the LR input.

3.2.3 Fourier Transforms in Neural Networks

Fourier transform is a powerful tool in the signal processing field, and it has also shown great potential in neural networks. Some initial practical applications aimed to use Fourier transform in neural networks to resolve signal processing problems. Gothwal et al. [83] and Mironovova and Bíla [84] combined Fast Fourier transforms (FFTs) and neural networks to identify cardiac arrhythmias disease. Zhang et al. [85] used FFTs to process vibration signals as features for training neural networks to classify fault and predict machine degradation in manufacturing systems.

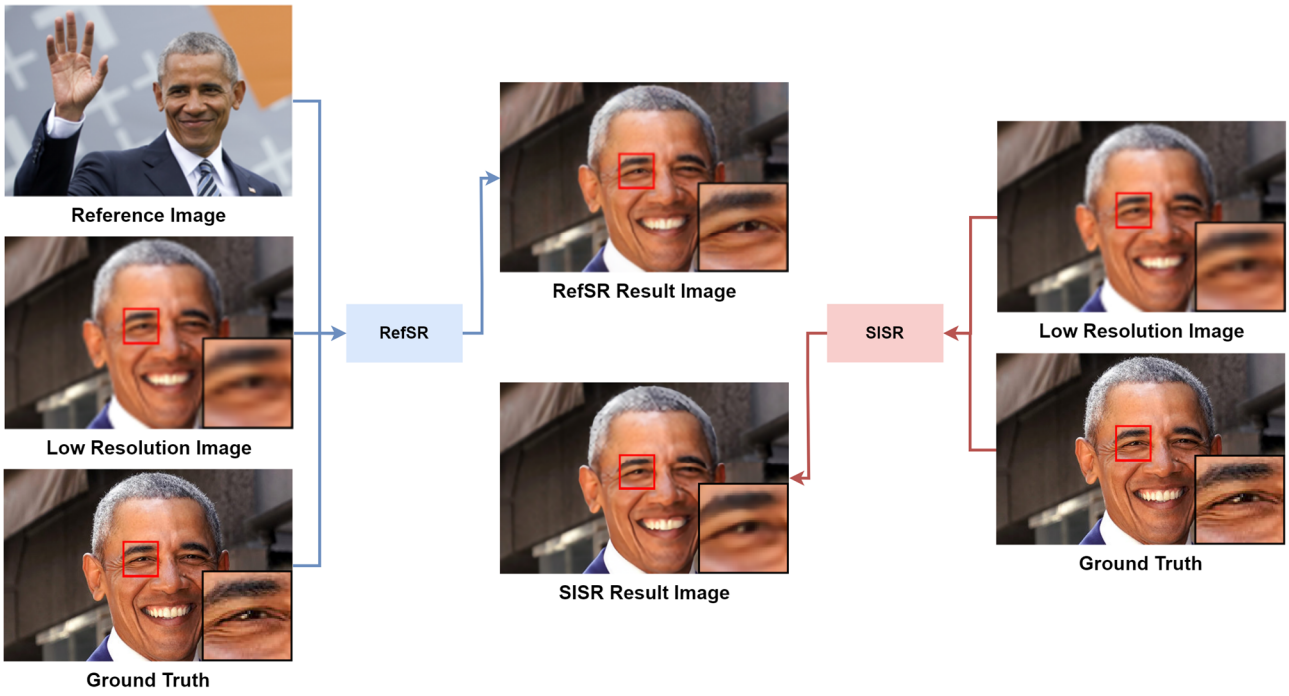

Figure 3-2: The input difference between a SISR [19] and RefSR [20] model during training (Images are from [20]). RefSR (Left) takes the LR, GT, and Ref images as input. SISR (Right) takes the LR and GT images as input. We can see from the results that RefSR generates better quality SR images because it can learn the high-resolution texture from Ref image.

As neural networks become deeper and deeper, more research focuses on utilizing Fourier transform with deep learning and CNNs. Fourier transform has been applied in CNNs to reduce computation time because multiplication in the frequency domain can be regarded as a convolution in the time or spatial domain. Mathieu et al. [86] proposed to compute convolutions as products in the Fourier domain, which speed up the training and inference speed significantly. Pratt et al. [87] proposed Fourier Convolutional Neural Networks (FCNN) for large computer vision tasks in which training is conducted entirely in the Fourier domain. Lin et al. [88] proposed a framework for Fast Fourier Transform-based deep neural network inference on embedded systems.

Rahaman et al. [89] adopted Fourier analysis that discovered empirical evidence, which shows that low frequency is learned first by the network, called the spectral bias. Xu et al. [90] proposed a novel learning-based frequency channel selection method that achieved good results on several tasks, including classification, detection, and segmentation. The proposed dynamic channel selection method also shows that the CNN models are more sensitive to low-frequency channels than high-frequency channels [90]. Such discoveries could indicate that CNN models are weaker at learning high-frequency details, which is challenging for super-resolution tasks. Li et al. [91] proposed the

first neural network super-resolution method solely by learning in the frequency domain. It shows advantages on the speed of the model with an imperceptible loss on results' quality.

In our work, we take a different approach to apply Fourier transform. We introduce to compute a loss function in frequency domain as a constraint for the neural networks instead of transferring the network itself into frequency domain.

3.2.4 Adaptation Problem for Super-Resolution

The adaptation problem has been a remaining issue for deep learning-based super-resolution methods. The problem is commonly seen when the testing data is not generated from HR images using the same downsampling kernel as the one used during training. SR datasets are usually created with paired HR and LR data, where LR data is downsampled from HR image with a known degradation method, usually a bicubic kernel. This approach is widely used to produce large training datasets but limits the performance of the trained model on testing LR images from unknown sources.

In recent years, researchers have worked on multiple ways to mitigate this issue. Zhang et al. [92] proposed to feed both LR image and its degradation maps as input to train with a set of downsampling kernels. Hussein et al. proposed adopting a correction filter to modify the input LR image to match the downsampling kernel used during training [93]. Dual Regression Networks (DRN) [15] adopted a closed-loop method that simultaneously learns the LR to HR mapping and the inverse mapping, as illustrated in Figure 3-3. The adaptation issue can be mitigated since the learning on HR to LR mapping can be regarded as estimating the underlying downsampling kernel [15].

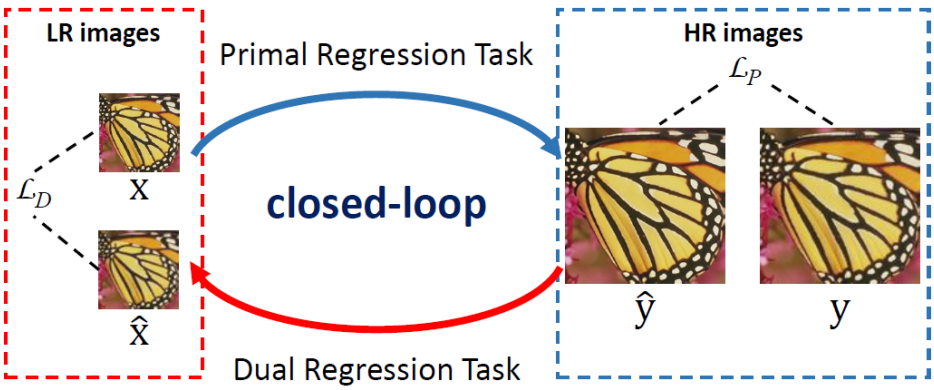

Figure 3-3: Dual regression training scheme used in [15]. The primal and dual regression form a closed-loop. The figure is reprinted from [15] with permission © 2021 IEEE.

The closed-loop method works well for a CNN-based SISR. We would like to explore if the idea would also work for RefSR, where the SR generation method is more complicated than a CNN model. Our proposed method is inspired by the closed-loop methodology and combines with training using multiple kernels.

Chapter 4. Synthetic X-ray Image Generation

4.1 Overview

Synthetic X-ray images are generated by simulating the X-ray image generation process based on CT data. The goal is to perform arbitrary view X-ray projection from CT data and calculate the X-ray attenuation coefficient. X-ray projection in real life is fundamentally a perspective projection where rays are emitted from the generator and travel through the human body. Our method description assumes an orthogonal synthetic X-ray image projection for better clarity and faster calculation in experiments.

We approach this issue from two aspects: projection and attenuation calculation. Our methods also contribute to these two problems. We rotate the CT data arrays with a matrix-based projection method to generate arbitrary view synthetic X-ray images. Then, we calculate the attenuation value using our custom-built lookup tables based on pixel values in the CT images. We introduce the details of these two techniques in Sections 4.2 and 4.3.

The general process of our method is shown in Figure 4-1. Assume our input has n number of CT slice images with the height of h and width of w. Firstly, in the data preprocessing period, images are resized as needed and load into two arrays in default order, $coordinates$ and $pixels$. This can be regarded as flattening the 3D CT arrays into two arrays, as shown in Figure 4-2. These two arrays save the CT data at pixel level. The length of them is $n * h * w$. $coordinates$ save each pixel's location information as $(n, h, w, 1)$. $pixels$ stores each corresponding pixel value in a 1D array. Secondly, in the data processing period, $coordinates$ will be multiplied with a 4 x 4 matrix M to transform the CT data according to the projection angles input. After this, we register the transformed coordinates $trancoordinates$ and $pixels$ into a dictionary $dict$. All the pixel values that share the same h and w in $trancoordinate$ will save into one key as (h, w) into the dictionary. Finally, we can use the created pixel value dictionary to calculate the pixel values in the resulting image. We use lookup tables to simulate different energy absorption levels of human organs and tissues and generate the final output synthetic X-ray image.

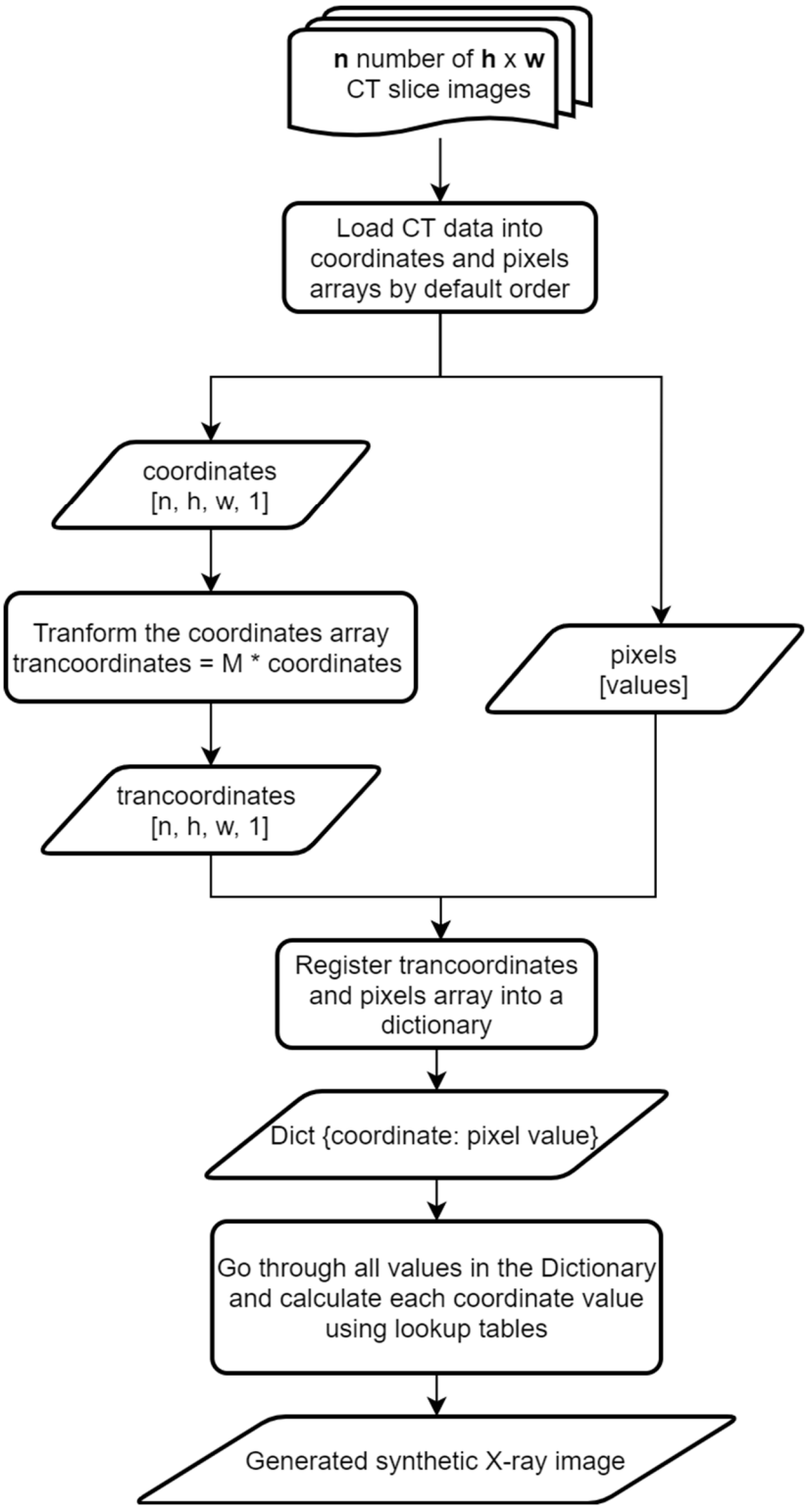

Figure 4-1: General process of the proposed method

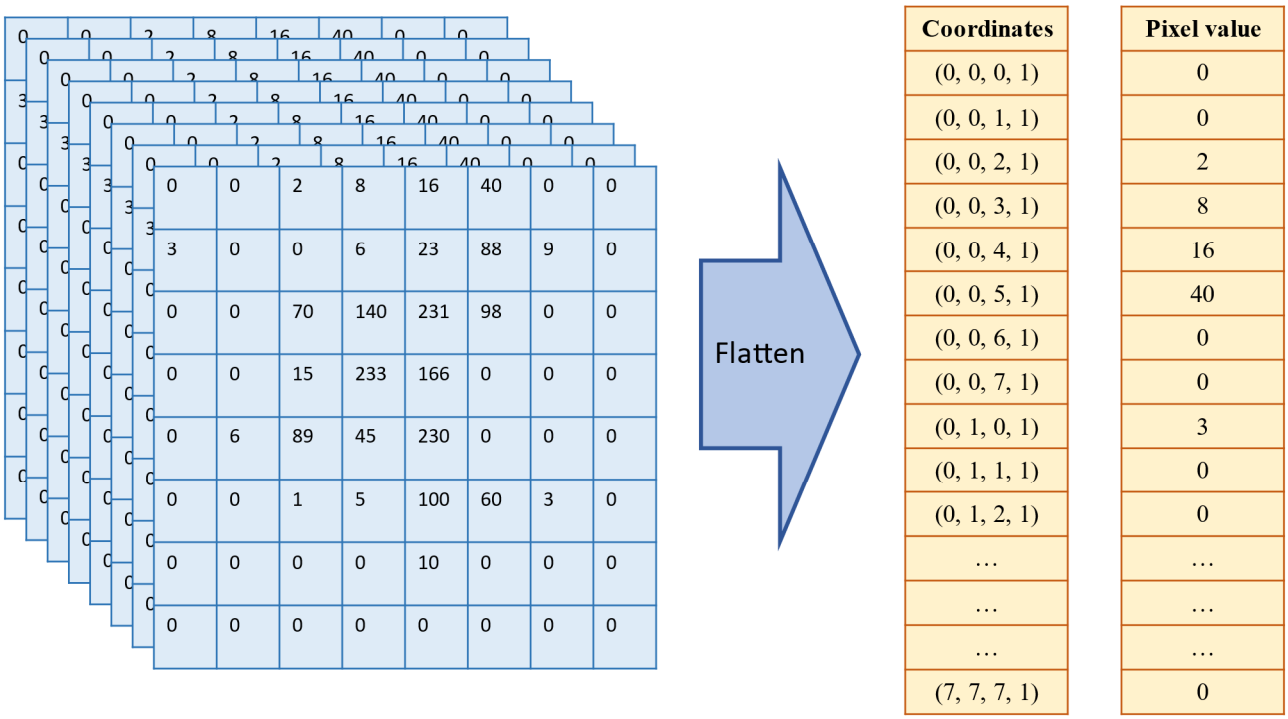

Figure 4-2: An example of 8 x 8 CT images in the data preprocessing period, where the CT 3D array is flattened.

4.2 Projection Method

We perform the projection first by a coordinate transformation and then by registering the new coordinates and pixels information into a dictionary. When loading the CT data coordinates, we take the number of CT images n as the axis z, h, and w as axis x and y. We need to transform the coordinates based on the input projection angles. Our projection method adopts matrix transformation instead of changing the projection viewpoint used in conventional ray-tracing methods. A comparison between the ray-tracing projection and ours is shown in Figure 4-3. The transformation matrices based on the projection angles for each axis is shown as follows:

$$M_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & 0 \\ 0 & \sin\theta_x & \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4-1}$$

$$M_y = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4-2}$$

$$M = M_x \cdot M_y \tag{4-3}$$

where θ_x and θ_y are the input projection angles for x and y axis. We can then calculate the final transformation matrix M using Equation 4-3. We can then get the $trancoordinates$ by computing the product of M and $coordinates$.

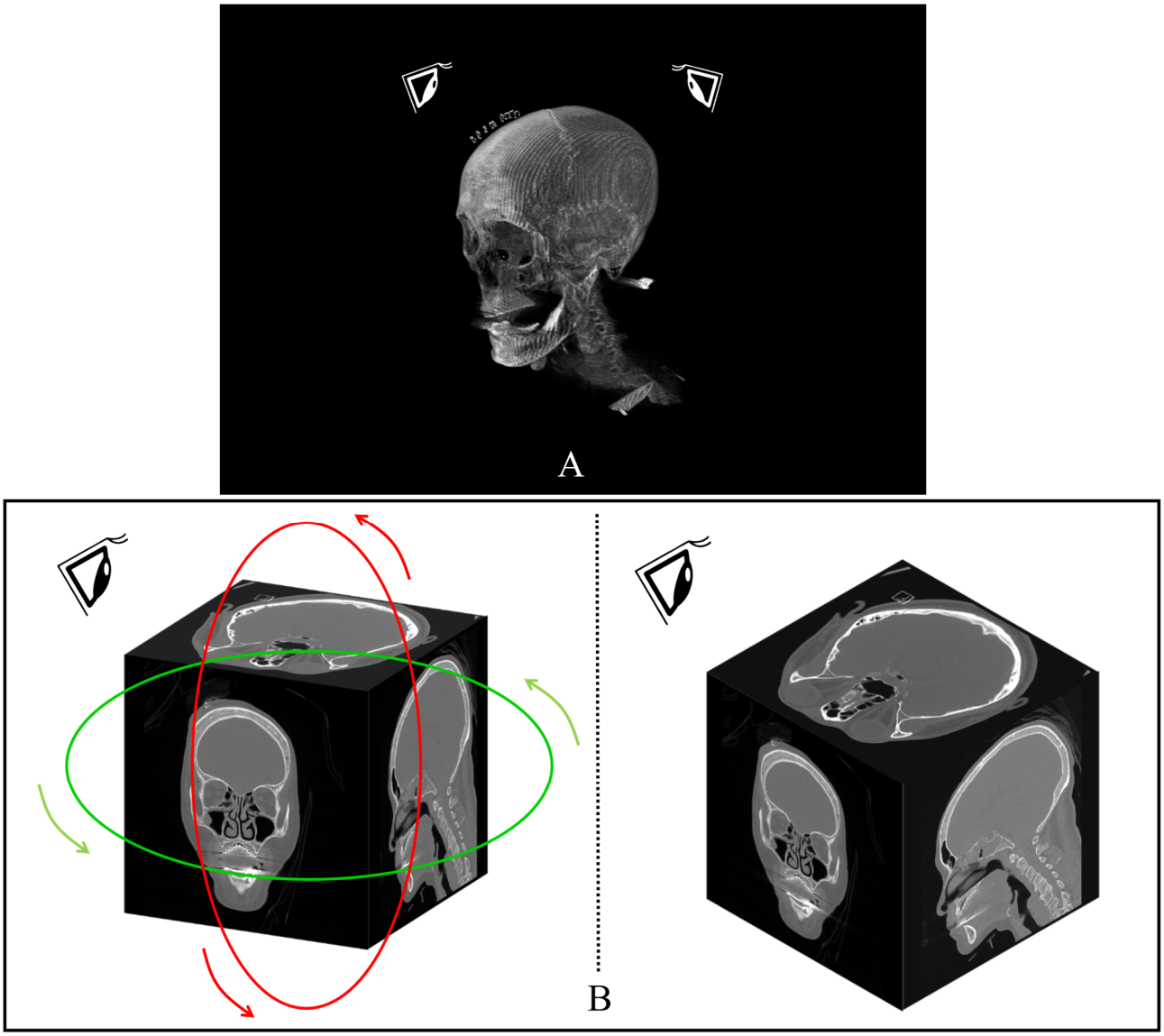

Figure 4-3: A comparison between the ray-tracing projection (A) and our matrix-based projection (B). The eye icon in the figures represents the point of view. The ray-tracing projection changes viewpoints to project X-ray from any angle of view. Our method fixes the viewpoint and rotates the CT 3D data.

After the transformation, the next step is to register all pixel coordinates and values into the dictionary. We illustrate the transformation and registering process in Figure 4-4. We save each new

coordinate (x, y) as a new key and the corresponding pixel value in the dictionary $dict$. We perform this by iterating over array $trancoordinates$ and $pixels$. The number of pixel values in each key varies after transformation based on different projection angles. For example, we have the same number of values in all keys if we store the original coordinates in the dictionary. And the keys on the periphery have a smaller density when we transform the coordinates. We show our algorithm to create the dictionary in Algorithm 1.

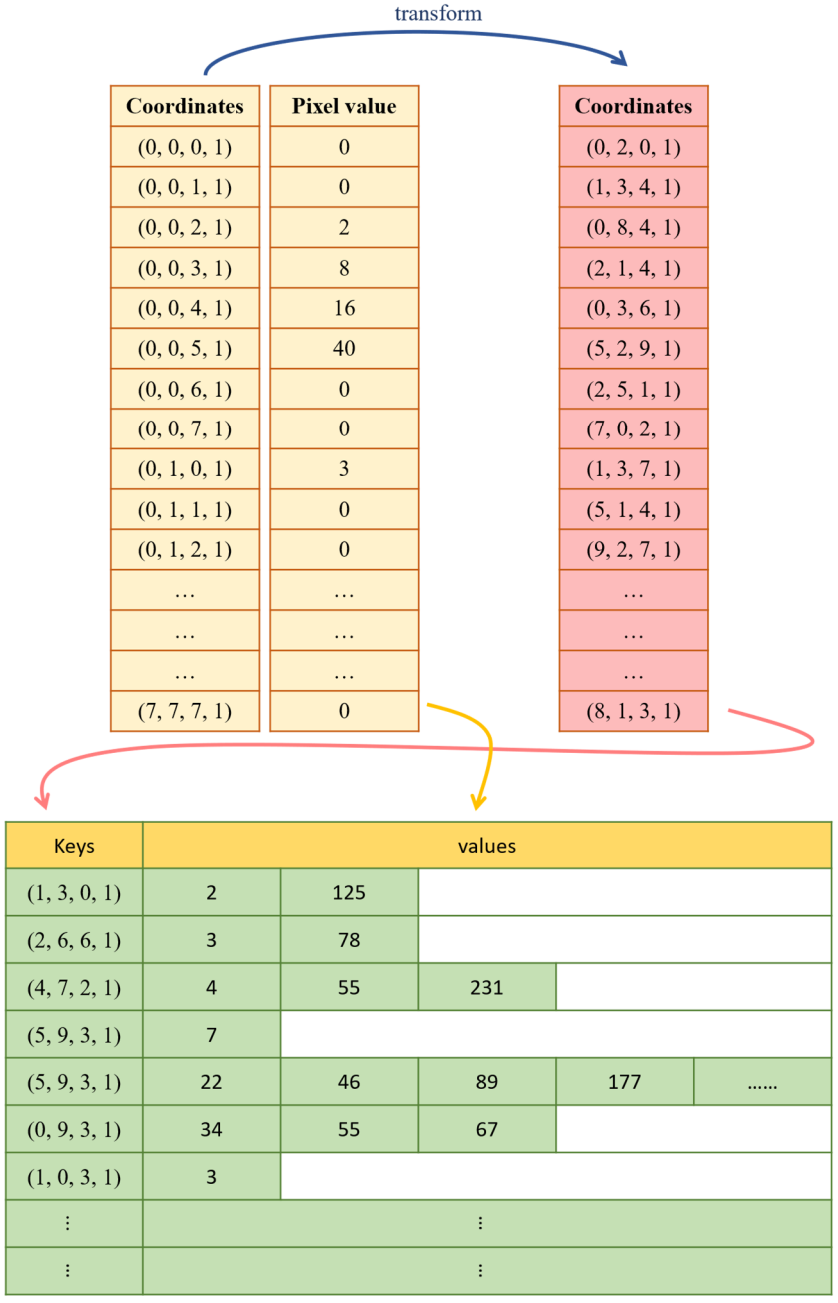

Figure 4-4: An illustration of the coordinates transformation and the dictionary creation process.

Algorithm 1: Algorithm for creating the dictionary

input : Arrays $trancoordinates$ and $pixels$
output: Dictionary $dict$
initialize an empty dictionary $dict$;
for $i \leftarrow 0$ **to** *length of* $pixels$ **do**
    extract x, y from $coordinates[i]$;
    $key = (x, y)$;
    if *key not in dict* **then**
        Create new key in $dict$;
        Store $pixels[i]$ value;
    else
        Store $pixels[i]$ value under $key(x, y)$ in $dict$;
    end
end

Algorithm 1: Algorithm for creating the dictionary from transformed coordinates and pixel values

4.3 Lookup Table

We build dynamic and multi-segment lookup tables to simulate the energy absorption of X-rays for CT head data. When X-rays travel through the human body, it is absorbed in different amounts by different tissues and organs. Directly using an average or a simple linear lookup table algorithm on the CT data can not simulate the characteristic of X-ray projection. We can segment different organs, tissues, and bones based on the Hounsfield unit in CT data or pixel values in grayscale images. We design our lookup table in six different segments with varying value thresholds based on the radiological density of different human tissues and computational experiments. We want to assign an absorption value to each pixel value (0 – 255). We define a penetration rate p to simulate the human body's absorption ratio of X-ray. This p is dynamically changing for each pixel coordinate along with the projection angle, which helps to adapt different density levels of CT volume for different pixel coordinates mentioned in the previous section. The constant parameter λ determines the brightness of the resulting image. It is set based on comparison to the brightness of real X-ray images. We set it to 3.6 by experimenting with the λ values in the range between 2.5 to 4.5. We compute p using the following equation, where d is the length of $values$ for each key in $dict$.

$$p = \lambda/d \tag{4-4}$$

The boundary and value threshold of each segment in the lookup table based on grayscale pixel values are shown in Table 3. We choose these values based on computational experiments and comparisons with corresponding real X-ray images. The general lookup table segments are estimated by considering the HU value in CT images, as introduced in Section 2.1.1. For example, air and bone are -1000 and 1000 HU in CT, and they correspond to 0 and 255 in grayscale images. Experiments start for segments with the smallest and largest pixel value in an image: air and bones. We can then gradually confirm the rest of the segments by comparing the results with the real X-ray image. We have one or two real X-ray images (SAG or COR view correspond to the view of CT data) paired with the CT data. The value threshold is set in a range from 0.05 to 1.00 only for testing the right segments at this point. We start to fine-tune the value threshold once the segments are confirmed. Similarly, we begin with the first and last segments' value thresholds and gradually confirmed the rest.

There is not a universal lookup table for all X-ray images. A lookup table can work well for X-ray images with different settings of the same body parts and organs with minor adjustments on the constant parameter λ. The lookup table would need adjustments in both segments and value thresholds for different body parts and organs.

Table 3: Lookup table segments, values threshold, and value segments

Lookup table segments	**Value threshold**	**Value segments**
[0, 74]	0.13	0 - 0.13
[75, 95]	0.25	0.13 - 0.25
[96, 140]	0.58	0.25 - 0.58
[141, 179]	0.83	0.58 - 0.83
[180, 210]	1.00	0.83 - 1.00
[211, 255]	1.17	1.00 - 1.17

To calculate the final lookup tables, we first get the value segments from value thresholds, as shown in Table 3. And then, each value in the lookup table is generated using its value segment. We build our dynamic lookup table by multiplying each value segment with penetration rate p. The

created lookup table LUT is a constant list with a length of 256. LUT is built by utilizing the $linspace$ function in python as follows:

$$LUT(x, y) = linspace(V_a \times p, V_b \times p, count) \tag{4-5}$$

$$count = |x - y| + 1 \tag{4-6}$$

where $linspace$ is a function that returns evenly spaced numbers over a specific interval. x and y are the start and end of the pixel segments. The previous and current segment's value thresholds are denoted as V_a and V_b. $count$ is the number of pixel values in the segments. For example, the LUT value of the first segment [0, 74] of the corresponding 75 pixels value is calculated as follows:

$$LUT(0, 75) = linspace(0 \times p, 0.13 \times p, 75) \tag{4-7}$$

We perform similar calculations for the following five segments until the entire lookup table is generated. The algorithm we use to calculate each pixel value for synthetic X-ray images from the lookup table is shown in Algorithm 2. We can generate the synthetic X-ray image by combining each calculated pixel value based on the coordinate information stored in the keys of $dict$.

Algorithm 2: Algorithm for calculating each pixel value in synthetic X-ray image from lookup table

input : $values$ of $dict$
output: each pixel value $sumval$
initialize the $sumval = 0$ to save calculated pixel value;
get d as the length of $values$;
for val *in* $values$ **do**
    $p = \lambda / d$;
    Create look up table LUT with p;
    $sumval = sumval + val * LUT[val]$;
end

Algorithm 2: Algorithm for calculating each pixel value in synthetic X-ray image from lookup tables

Chapter 5. Synthetic X-ray Super-Resolution

5.1 Overview

This section introduces the improvements we made for synthetic X-ray image super-resolution in two aspects. We present the frequency domain loss for super-resolution in 5.2. Then, we describe our adaptation network for super-resolution in 5.3. Please refer to Table 1 for super-resolution abbreviations and their explanations for the abbreviations used in this chapter.

5.2 Frequency Domain Loss for Super-Resolution

We aim to reduce the artifacts generated by the network while not losing the fine details. To achieve this, we introduce frequency domain loss as illustrated in Figure 5-1. Firstly, we explore the frequency domain components for HR image, LR image, and resulting images from different methods in the magnitude spectrum. Then, we introduce our loss function computed in the frequency domain.

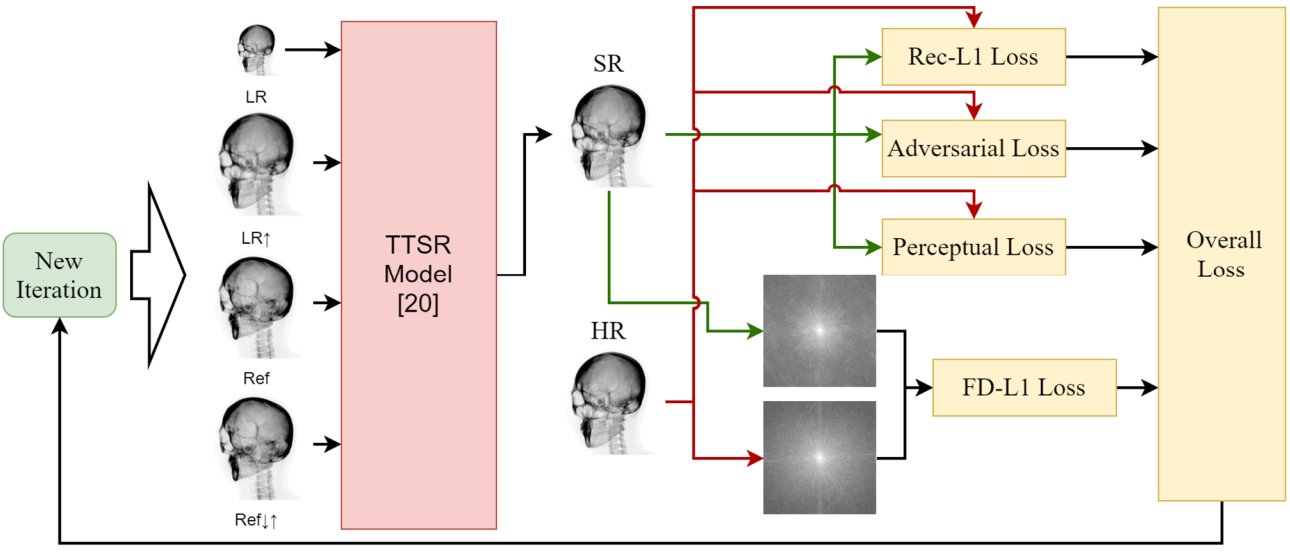

Figure 5-1: Our proposed RefSR method, TTSR-FD. We add the frequency domain loss on the TTSR [20] model. The four input images are low-resolution (LR), up-sampled LR, reference images (Ref) and down/up-sampled Ref images. The super-resolution image (SR) is the model output. The high-resolution image (HR) is the ground truth image.

5.2.1 Magnitude Spectrum for Images

We adopt Fourier transform to reveal frequency patterns that are not visible in the spatial domain. In Figure 5-2, we show a few examples of spatial domain images and magnitude spectrums (MS) of HR and LR synthetic X-ray images. The magnitude of an image in frequency domain is introduced in 2.9.2. We can observe similar white areas in the center of both HR and LR MS images. This shows that LR images preserve most of the low-frequency details. On the other hand, much information representing high-frequency details is lost, as shown in the periphery of the LR MS image. These are the information we want to recover with super-resolution methods.

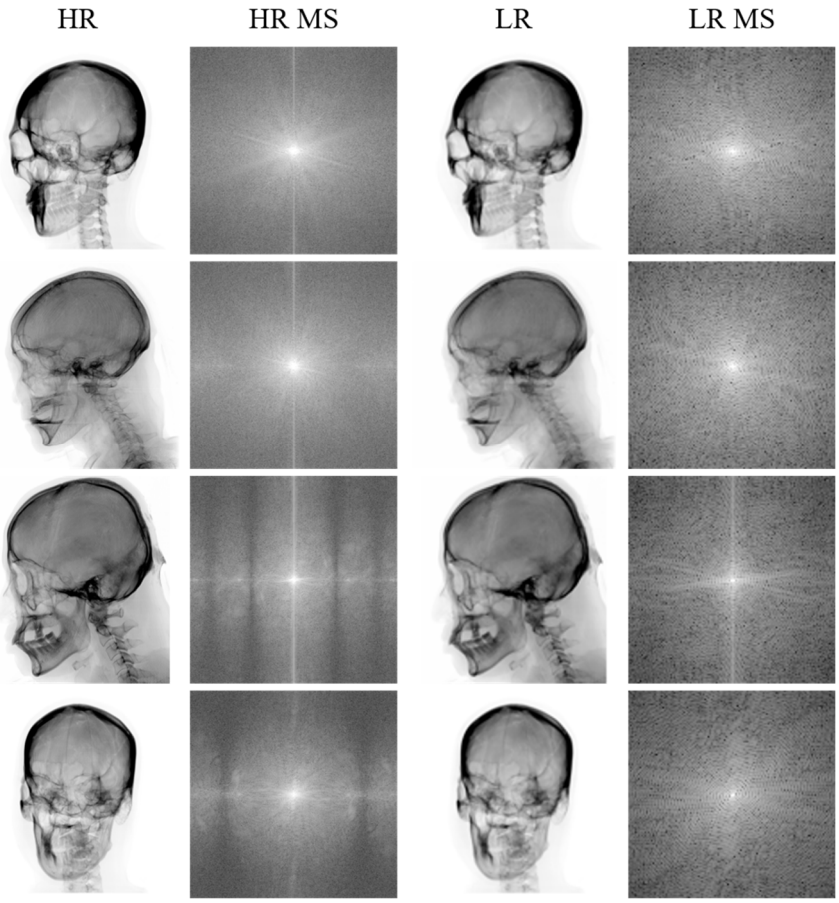

Figure 5-2: A few examples of magnitudes of synthetic X-ray images between HR and LR images. HR images are 512 x 512 resolution. LR images are 128 x 128 resolution. The first and third columns are spatial domain images. The second and fourth columns show the magnitude spectrum (MS).

We also want to see how well upsampling and super-resolution methods can restore images in the frequency domain. We compare the HR ground truth image, input LR image, bicubic unsampled

image, and several results of well-known super-resolution methods such as RCAN, ESRGAN, and TTSR. Their spatial domain images and magnitude spectrums are shown in Figure 5-3. While the bicubic unsampled image shows the worst results, we observe that all super-resolution models can not learn adequate high-frequency details. We also present our TTSR-FD result as a comparison of the improvement.

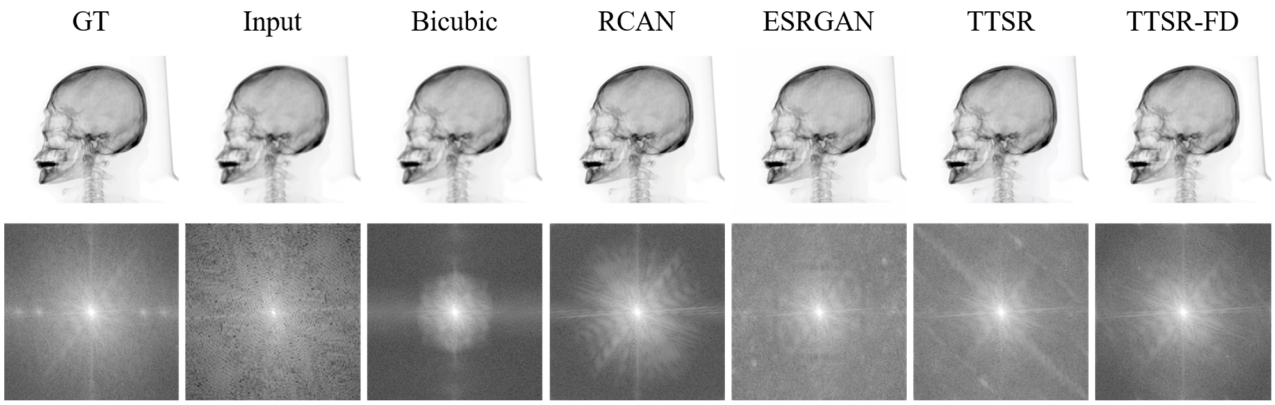

Figure 5-3: Images in the magnitude spectrum where the first is the ground truth (GT) image. The second is the low-resolution input image, and the last TTSR-FD is our result.

In general, we have whiter regions in the center, indicating that all images contain more low-frequency contents than high-frequency ones. We can see from the input image spectrum that it lost lots of high-frequency details compared to the GT image. And the bicubic method is not able to generate high-frequency details. RCAN could only generate limited high-frequency details. ESRGAN can learn some good high-frequency details and shows sharp details in the spatial domain. But some odd patterns are shown in frequency domain, such as a rectangle shape contour in the center. This indicates that it generates artifacts in the resulting image. TTSR can generate fine high-frequency details but also fails to match some patterns. Our TTSR-FD presents the best similarity in the frequency domain compared to the GT image. We can find out that the frequency domain contains lots of information that is not shown in the spatial domain.

5.2.2 Frequency Domain Loss

Based on the previous section's observation, we hypothesize that a loss function computed in the frequency domain can increase the resulting image quality of RefSR. We aim to use this loss to set up a constraint during training that can force the model to learn more from the data and generate fewer artifacts and noise. We choose to build this constraint with a pixel-wise loss function to ensure

the network learns from the image in frequency domain. We utilize $L1$ loss, which is proven effective for super-resolution, also more robust and easier to converge compared to $L2$ loss. Our network architecture is shown in Figure 5-1. We first transfer model output and ground truth images of each iteration into the frequency domain. Then, we compute their $L1$ loss, and feed back to the network. We apply this loss to our baseline model TTSR [20], which has three loss functions. We partially rewrite this part from Section 2.7.1 for reading convenience. The overall loss of the baseline model is:

$$\mathcal{L}_{overall} = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{per}\mathcal{L}_{per} \tag{5-1}$$

where $\mathcal{L}_{rec}$ is the reconstruction loss with L_1 loss. $\mathcal{L}_{adv}$ is the adversarial loss using WGAN-GP [58]. $\mathcal{L}_{per}$ is the perceptual loss that includes a normal perceptual loss and a texture wise loss [20]. λ is the weight coefficients for the loss functions. Below is the overall loss of our model.

$$\mathcal{L}_{overall} = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{fd}\mathcal{L}_{fd} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{per}\mathcal{L}_{per} \tag{5-2}$$

We add a frequency domain loss $\mathcal{L}_{fd}$ to improve the network performance. In each iteration, a batch of SR and HR images are transferred into the frequency domain, and then their $L1$ loss is calculated.

$$S^{HR} = f_{rfft}(I^{HR}), \quad S^{SR} = f_{rfft}(I^{SR}) \tag{5-3}$$

$$\mathcal{L}_{fd} = \frac{1}{CHW}\|S^{HR} - S^{SR}\|_1 \tag{5-4}$$

where I^{HR} is the high-resolution ground truth image, and I^{SR} is the predicted super-resolution image from the model output. S^{HR} and S^{SR} are the corresponding spectrum images after Fourier transform. C, H, W are the channel, height, and width of the HR image. f_{rfft} represents the FFT function from PyTorch that takes real-valued input. The built-in FFT function from PyTorch can process tensors directly without the need for data type transferring, thus improving computation efficiency. There is no significant increase in the time complexity when using frequency domain loss during training.

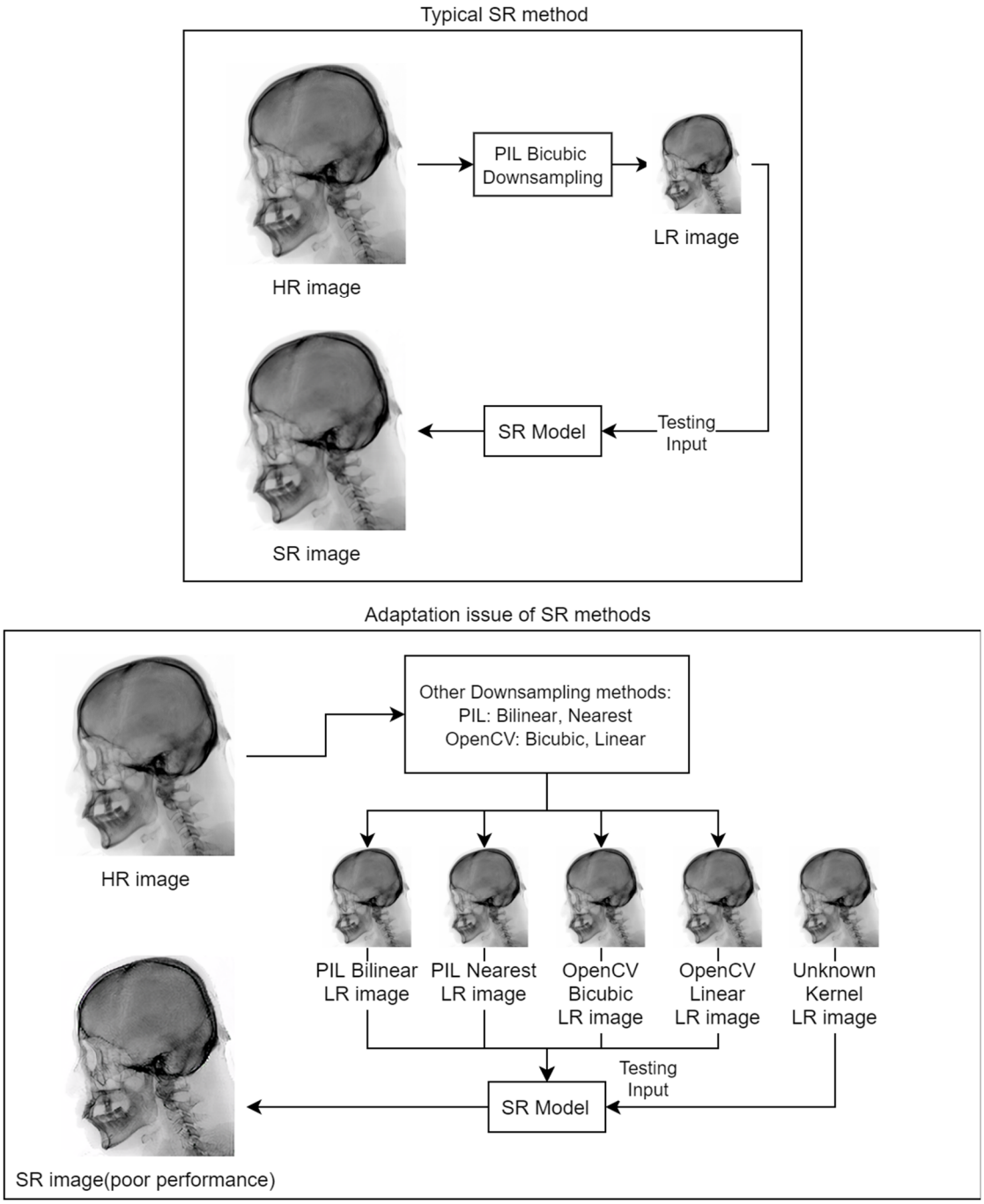

Figure 5-4: Adaptation issue in super-resolution field. Up figure: the conventional SR method testing process produces the testing images using the same kernels that created the training data. Down figure: In the adaptation issue scenario, the testing images are produced with kernels that are not used to create the dataset or unknown kernels.

5.3 Adaptation Network for Super-Resolution

In this section, we aim to train a more general model for testing images created from kernels that are not used during training. We illustrate the adaptation issue of super-resolution methods in Figure 5-4. To achieve our goal, we propose an adaptation network that makes two modifications to the TTSR architecture. Firstly, we randomly downsample the HR images to LR images using multiple kernels while loading the dataset. Secondly, we train an additional dual model that forms a closed-loop with the main model. And we add the dual regression loss to the overall loss while training. The architecture of our proposed model is shown in Figure 5-5.

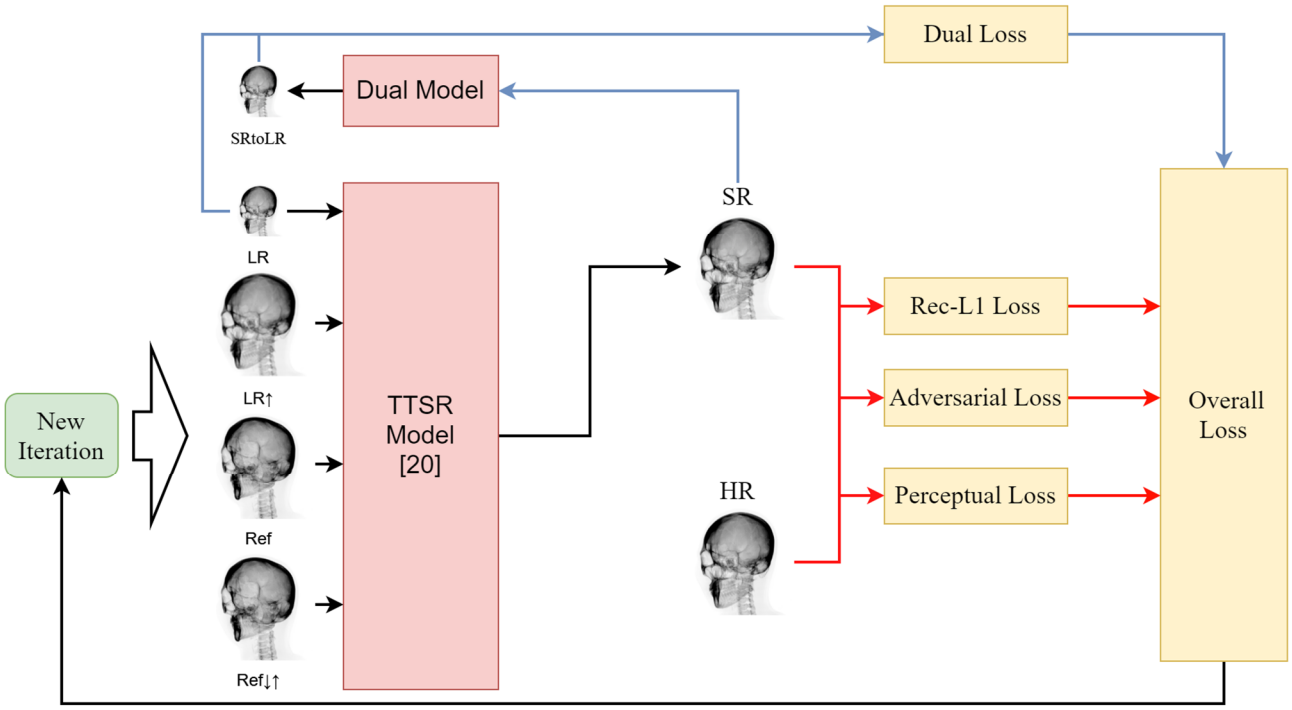

Figure 5-5: Our proposed architecture. We add a dual model and form a closed-loop with the main TTSR [20] model. In this way, the model can learn the reverse mapping from the model output SR to the input LR image. The dual regression loss is then added to the overall loss.

5.3.1 Differences in Different Kernels

To understand why changing degradation methods used in testing yields a significant performance decrease for the existing SR method, we first study the differences among different downsampling kernels. We show the differences in images by calculating the absolute pixel value

difference of the two images. The calculated figure is brighter if the two images are very different and darker if the two images are similar. We also transfer the images into the frequency domain to see differences for high and low-frequency details. We choose two kernels from the PIL (Python Imaging Library) package and two kernels from the OpenCV package to downsample the same image and then compare them with PIL Bicubic downsampled result, as shown in Figure 5-6. We can observe the significant differences among these five kernels. PIL Bilinear is the closest to PIL Bicubic among the four kernels compared here. This is also reflected in our results section for the adaptation problem in Section 6.4.2.1, where testing images produced with PIL Bilinear gained better performance without using the adaptation network.

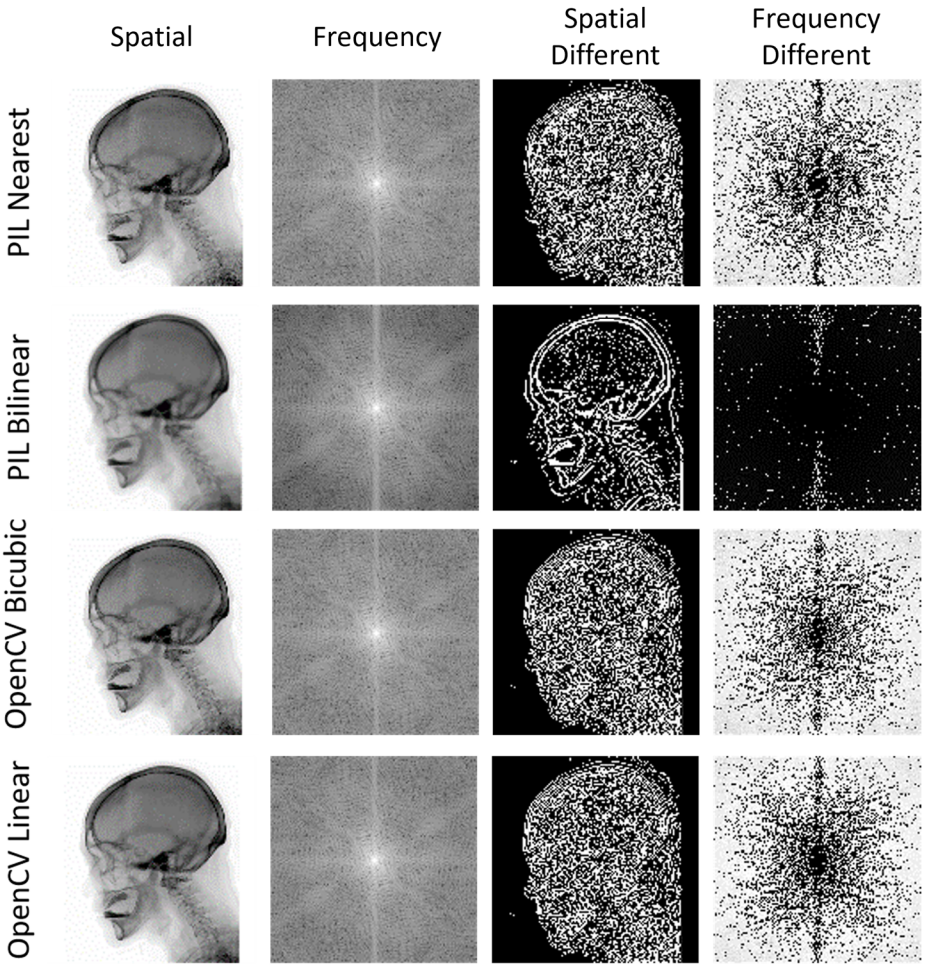

Figure 5-6: The difference between PIL Nearest, PIL Bilinear, OpenCV Bicubic, OpenCV Linear with PIL Bicubic in spatial domain and frequency domain. The first and second columns show the spatial and frequency domain of the images. The third and fourth columns show the differences between the images' spatial and frequency domain in absolute pixel value differences.

5.3.2 Learning Multiple Kernels

Most super-resolution methods use only Bicubic downsampling kernel during training and show poor performance when the testing LR image is not downsampled with the same kernel. In our proposed method, three different downsampling kernels are randomly applied to generate the LR images. We use the Bicubic, Bilinear, and Nearest from Python Imaging Library (PIL) in our training process. Learning multiple kernels enables our model to be more generalized. Our model achieves better performance for testing images produced with kernels either used or not used in training.

5.3.3 Dual Model with a Closed-Loop

The potential mapping space is increased due to more kernels being used in training. This causes the model to become unstable and harder to train. We adopt the closed-loop method [15] to address this issue and use a dual regression loss to increase our model performance. The goal is to train a model to predict the origin LR input image from the generated HR image. Essentially, the dual model is trying to estimate downsampling kernels. We propose to use a simple downsampling model to learn the inverse mapping from HR to LR. The network structure of our dual model is shown in Figure 5-7. It consists of two downsampling blocks build with a few Conv2D operations and LeakyReLU activation layers. The stride number is symbolized as $S1$ or $S2$. This dual model is suitable for the X4 super-resolution task used in our work. The model can be modified for X2 or X8 tasks by removing or adding layers. For example, the X8 task would require four downsampling blocks. We use $L1$ loss for our dual regression loss, follow the same as in [15]. This loss adds to the TTSR overall loss during each iteration. The dual regression loss is defined as follows:

$$\mathcal{L}_{dual} = \frac{1}{CHW} \| I^{LR} - D(I^{SR}) \|_1 \tag{5-5}$$

where I^{LR} and I^{SR} are LR and predicted HR images from the TTSR model. $D(I^{SR})$ represents the predicted LR image produced by the dual model. C, H, W are the channel, height, and width of the LR image. With the dual regression loss, the overall loss of the model is as follows:

$$\mathcal{L}_{overall} = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{per}\mathcal{L}_{per} + \lambda_{dual}\mathcal{L}_{dual} \tag{5-6}$$

where λ_{dual} is the weight coefficient of the dual regression loss. The other loss functions are similar in Section 5.2.2. $\mathcal{L}_{rec}$ is the reconstruction loss with L_1 loss. $\mathcal{L}_{adv}$ is the adversarial loss using

WGAN-GP [58]. $\mathcal{L}_{per}$ is the perceptual loss that includes a normal perceptual loss and a texture wise loss [20].

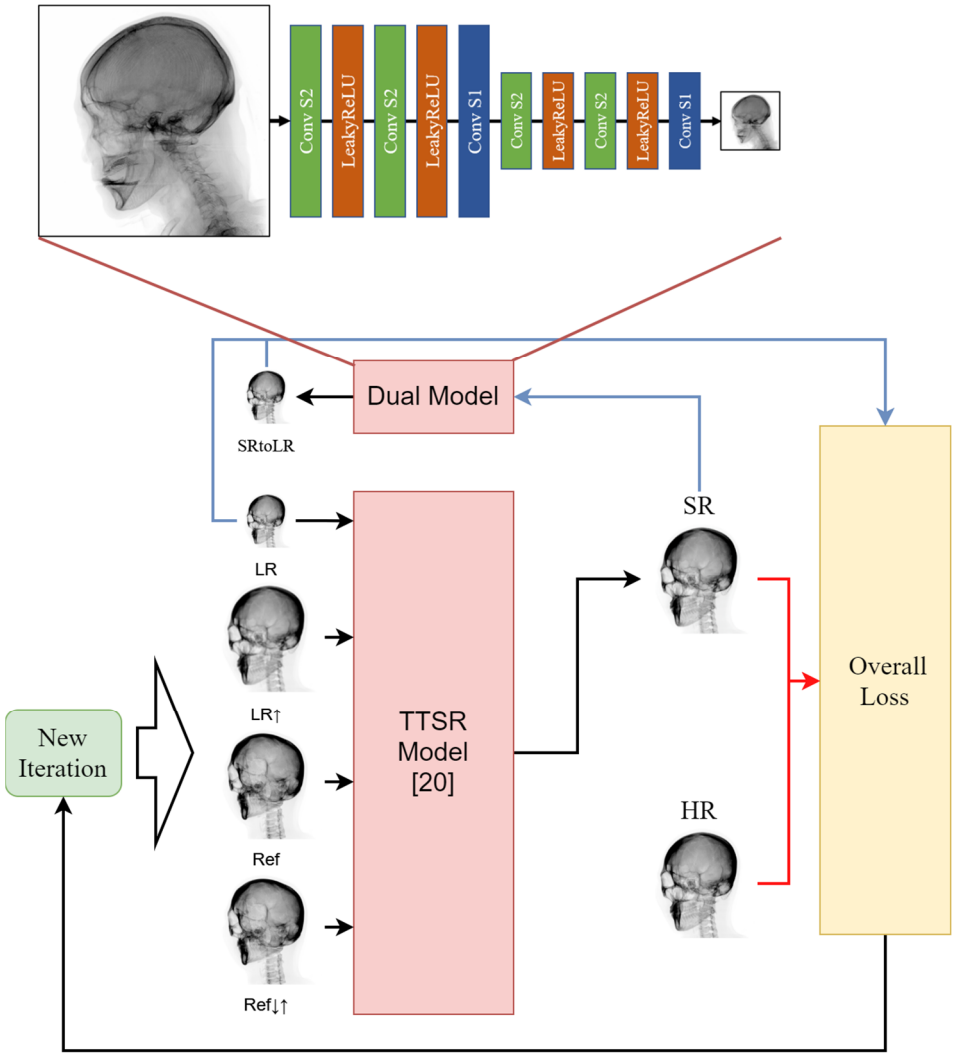

Figure 5-7: Dual Model used in our method. It consists of two downsampling blocks and forms a closed-loop with the main model. The downsampling block is built with a few Conv2D operations and LeakyReLU activation layers. $S1$ or $S2$ represents the stride number.

Chapter 6. Experiments

In this chapter, we introduce the experiments for both synthetic X-ray image generation and super-resolution. Firstly, we explain the implementation of the synthetic X-ray image generation method and show our results. Secondly, we introduce the datasets used for super-resolution experiments and how we build them. These include a RefSR dataset created with our synthetic X-ray image generation method, a real X-ray RefSR dataset created with real chest X-ray images, and a synthetic X-ray image SISR version dataset. Thirdly, we introduce the setup of our experiment for synthetic X-ray image super-resolution. Then, the results of our super-resolution method on synthetic X-ray images and real X-ray images are given. An ablation study is also conducted on the impact of using frequency domain loss on synthetic X-ray image super-resolution. After this, we introduce our results on the adaptation network. Finally, we discuss the results of our experiments in general and the impact of our methods in the field.

6.1 Synthetic X-ray Image Generation

6.1.1 Implementation Details

We experiment with our method on real human head CT data. Our data is provided by Dr. Jae Chul Koh from Korea University Anam Hospital. It contains seven different head CT series from five patients with Sagittal (SAG) and Coronal (COR) views. The number of slices in these series ranges from 48 to 145, and each slice resolution is 512x512. We generate arbitrary view synthetic X-rays images for each CT series. Python is used to implement our algorithm. We do not utilize any GPU resources to generate our results. Instead, we use Numba [94] package to speed up our algorithm. Numba can translate python functions to optimized machine codes, enabling python to reach the C language's speed, especially for numerical calculations. We use linear interpolation to improve image quality based on the number of CT slices in each series and the DICOM file attributes such as slice thickness.

We compute our results using Python implementation, but we also build a C# version of our code in Unity3D. Because of the simplicity of our method, we can implement our algorithm in

Unity3D without any third-party packages. The model performance is similar to the implementation using Python with Numba. We show more details of this part in Chapter 7.

6.1.2 Results

Evaluating synthetic X-ray image results could be a challenging task. It is difficult to measure with metrics as we do not have ground truth images. Instead, we perform some visual analysis for our results. We show our results generated from CT data of one patient in the range from -40º to 40º with a gap of 20º for both x and y-axis in Figure 6-1. We observe overall improved image quality for our results compared to conventional methods. However, the quality reduces when the rotation angle becomes far away from the original viewpoint of the input CT series.

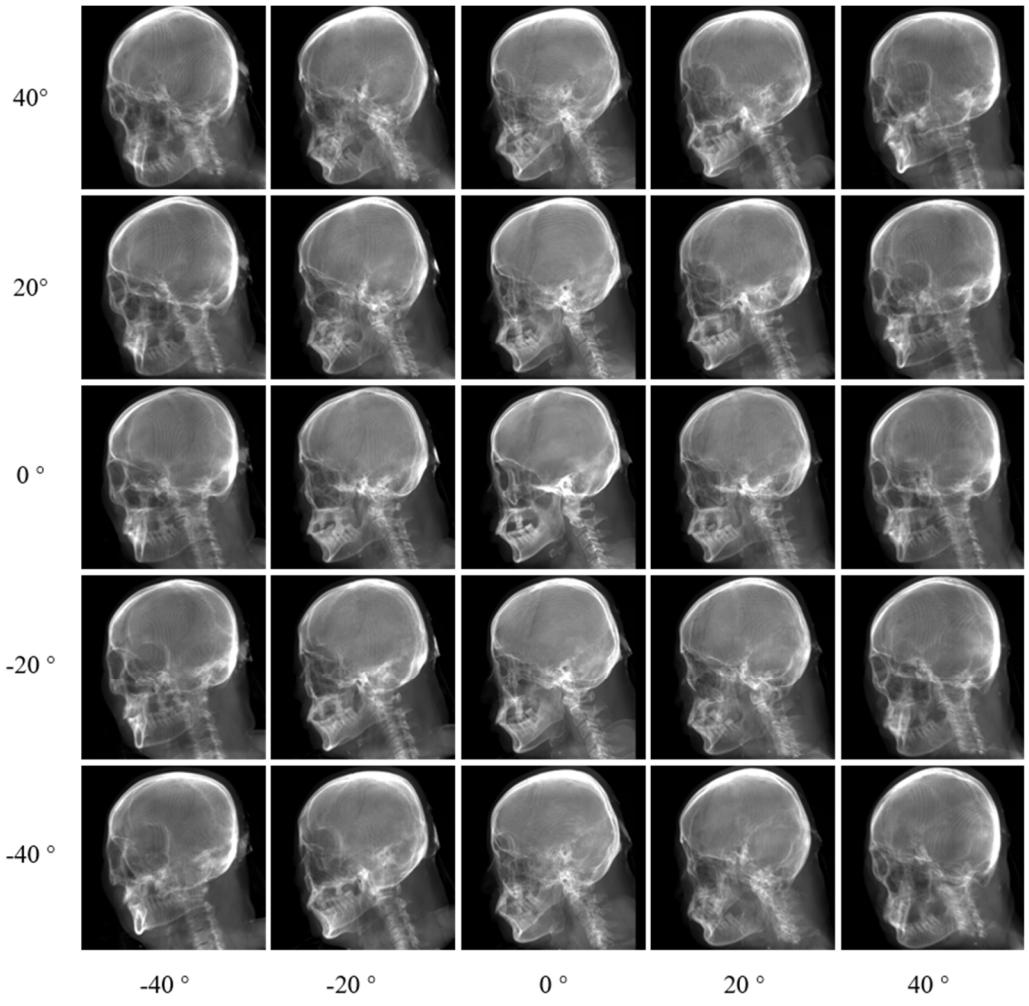

Figure 6-1: Results of generating view from -40º to 40º with a gap of 20º in both x and y axis for one patient.

We show an extreme situation where the projection angle is 90º in two axes (Coronal view and Axial view), as shown in Figure 6-2. This is one of the most challenging issues for generating synthetic X-ray images from CT slices. We can mitigate this issue by using multi-view CT data, but they are not always available. Figure 6-5 provides more synthetic X-ray images of different patients in random projection angles to show that our lookup table parameters work for CT head data of different patients.

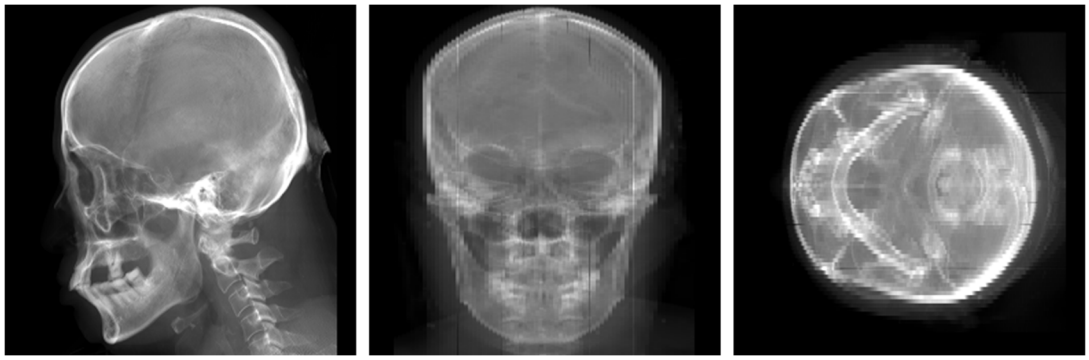

Figure 6-2: Results of generating in Saggital views, Coronal view and Axial view. The input CT data is from the Sagittal view.

We compare our results with a conventional ray-tracing method, as shown in Figure 6-3. Two patients are included in the comparison. We show three images for each patient from 0º to 40º with a gap of 20º. We use MeVisLab [13] to generate the ray-tracing results with Siddon's ray tracing [10] method. We can observe from the comparison that our results have better contrast and richer content. This is because conventional methods are often used for registration purposes and focus more on structural position, which can work fine for bones but not for soft tissues. Benefits from our dedicated multi-segments lookup tables, our generation method could produce fine details for tissues and bones. Our method also shows good quality on different tissues instead of focus on reconstructing bone structures. A direct comparison between our results and results of other methods focusing on chest CT to X-ray DRRs is shown in Figure 6-4. Overall, our results provide a more detailed and much better contrast image compared to conventional methods.

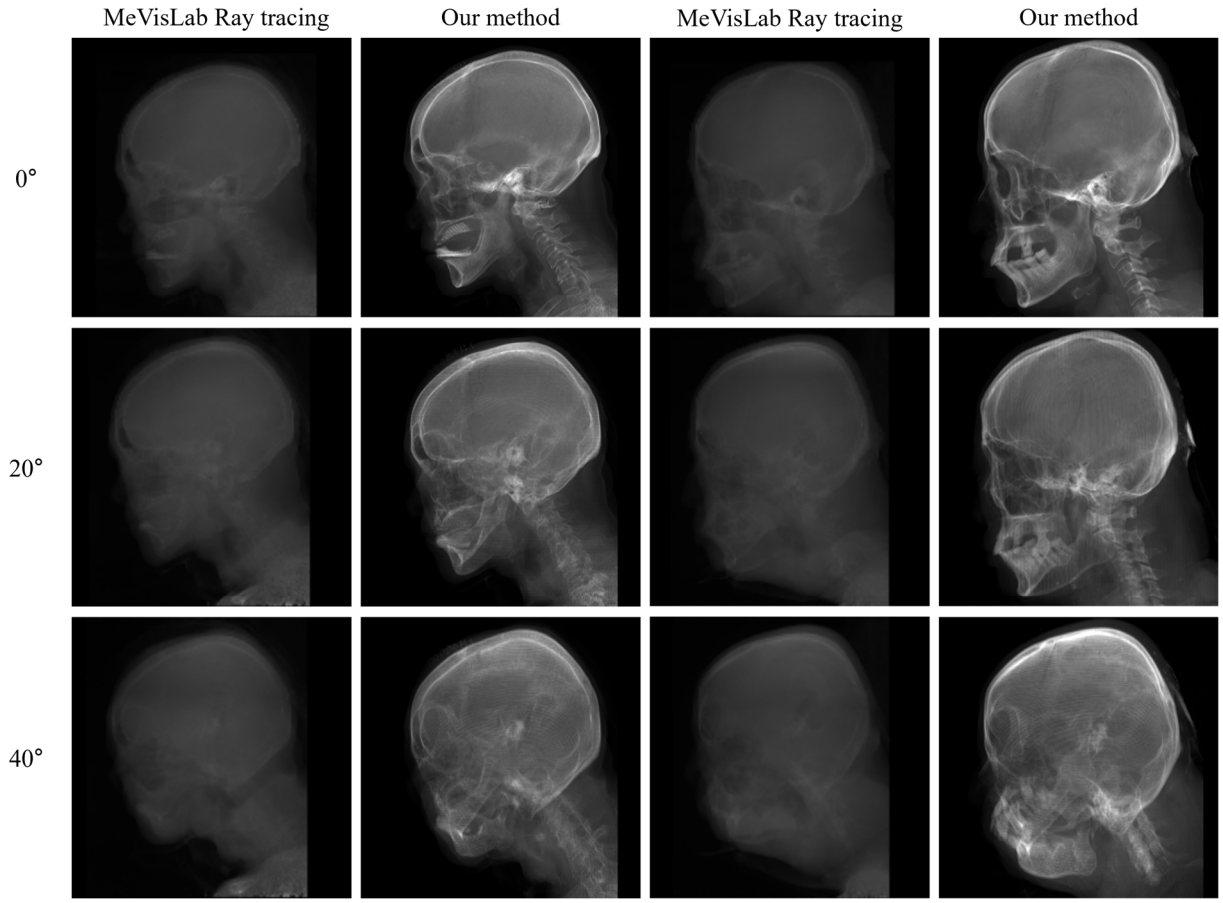

Figure 6-3: Visual comparison of our method with a ray-tracing method. We include synthetic X-ray images generated from 0º to 40º with a gap of 20º for two patients. Our results show better quality and more refined details compared to results from publically available MeVisLab software.

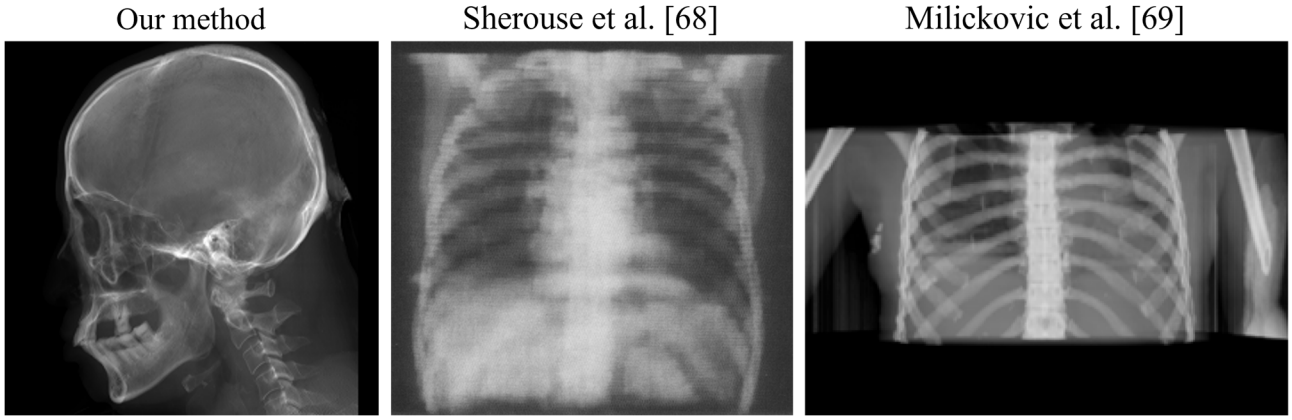

Figure 6-4: A comparison of results between our method and two previous approaches [68] [69] worked on CT to X-ray image reconstruction. Our method shows better image quality in general.

The average computation time performance for different input CT data resolutions is shown in Table 4. It is worth mentioning that the computation time for each generation scenario varies according to the projection angle. We observe that most of the time is used to create the dictionary during the data processing part. The computation time is not minimal as the algorithm aims to generate higher quality images compared to fast ray-tracing methods. The significant increase of computation time for high-resolution synthetic X-ray image generation time is also one of the motivations of our research in synthetic X-ray image super-resolution. Our goal is to use a super-resolution method to reconstruct a high-resolution image from a low-resolution synthetic X-ray image. Thus we can produce high resolution and good quality synthetic X-ray images in a much shorter time.

Table 4: Computation Time Performance

Input CT Resolution	Computation Time (s)
128 x 128	6.3 s
256 x 256	18.3 s
512 x 512	111.4 s

Figure 6-5: More generation results from other patients in random projection angles.

6.2 Dataset for Super-Resolution

We create two RefSR datasets to evaluate our method, a synthetic head X-ray images dataset and a real chest X-ray images dataset. Both datasets are constructed following similar approaches as described in [57] while making some adjustments considering the characteristic of synthetic and real X-ray images.

6.2.1 Synthetic X-ray Image Dataset for RefSR

This section introduces the generation process of our synthetic X-ray image dataset for reference-based super-resolution. We use the same CT data as used in Section 6.1. The data description is partially rewritten for reading convenience. Dr. Jae Chul Koh from Korea University Anam Hospital provides the head CT data. It contains seven different head CT series from five patients with Sagittal (SAG) and Coronal (COR) views. The number of slices in these series ranges from 48 to 145. The resolution of each slice is 512 x 512. We show more details of the CT data properties in Table 5. We used linear interpolation with different variables to fill the different gap sizes between slices for each series during the pre-processing period. The image interpolation number between two CT slices for generating 512x512 resolution synthetic X-ray images is also shown in Table 5.

We followed a similar approach described in SRNTT [57] to generate our dataset while making some adjustments considering the characteristic of synthetic X-ray images. The dataset proposed in [57] is constructed based on the CUFED [95] dataset, which contains 1,883 albums from diverse events in daily life. Images numbers in each album range from 30 to 100 images. The images are separated into different pairs based on SIFT [96] feature matching. Similarity levels are also ranked based on the number of SIFT features' best matches. Image pairs are then randomly cropped into five 160x160 patches for both HR and Ref images. There are a total of 13761 paired patches in the CUFED5 training set. 126 groups of samples are in the testing set. Each group has one HR image and four Ref images in different similarity levels. In Figure 6-6, we show two examples of testing groups from the CUFED5 testing set.

Figure 6-6: Examples from the CUFED5 testing set. From left to right are the HR image and the corresponding Ref images. The similarity levels are L1, L2, L3, and L4, respectively.

Table 5: CT data properties

Patient number	View Planes	Slices number	Resolution	Interpolation Number
Patient 1	COR	67	512 x 512	4
Patient 2	COR	48	512 x 512	5
Patient 2	SAG	60	512 x 512	4
Patient 3	SAG	145	512 x 512	1
Patient 4	SAG	74	512 x 512	5
Patient 5	COR	96	512 x 512	2
Patient 5	SAG	80	512 x 512	3

Our synthetic X-ray image dataset uses synthetic X-ray images for fluoroscopic images that differ from the results shown in Section 6.1.2, where the bone is in black instead of white color. This is because the color value is reversed for X-ray images used in C-armed machines. The training set consists of 225 (15x15) input images for each CT series. They are generated with a sampling step of 24 degrees for both the x-axis and y-axis. We use synthetic X-ray images generated from different angles as reference images. So the corresponding reference image is generated with a random projection angle from -45 to 45 degrees based on the projection angle of the input image. Each input and reference image are then randomly cropped into five patches of 160x160. We build a large training set SynXray_L consists of 1350 synthetic X-ray image pairs with 512 x 512 resolution from

four out of five patients. We also create a small dataset SynXray_S consists of 410 synthetic X-ray image pairs randomly selected from four out of five patients. After cropping into patches, the SynXray_S has 2050 paired patches, and SynXray_L has 6750 paired patches. We leave one patient out during training to ensure our testing data is not seen during training. The testing set X-ray images are randomly generated with a sampling step of 20 degrees. There are 30 groups of paired images in the testing set from all five patients. This setting makes sure we have images from different projection angles compared to the training data and guarantees we always have test data from a patient not in the training set. The similarity levels of the images in each pair are ranked based on the differences in angles. We show two testing group examples from our synthetic X-ray image RefSR dataset in Figure 6-7.

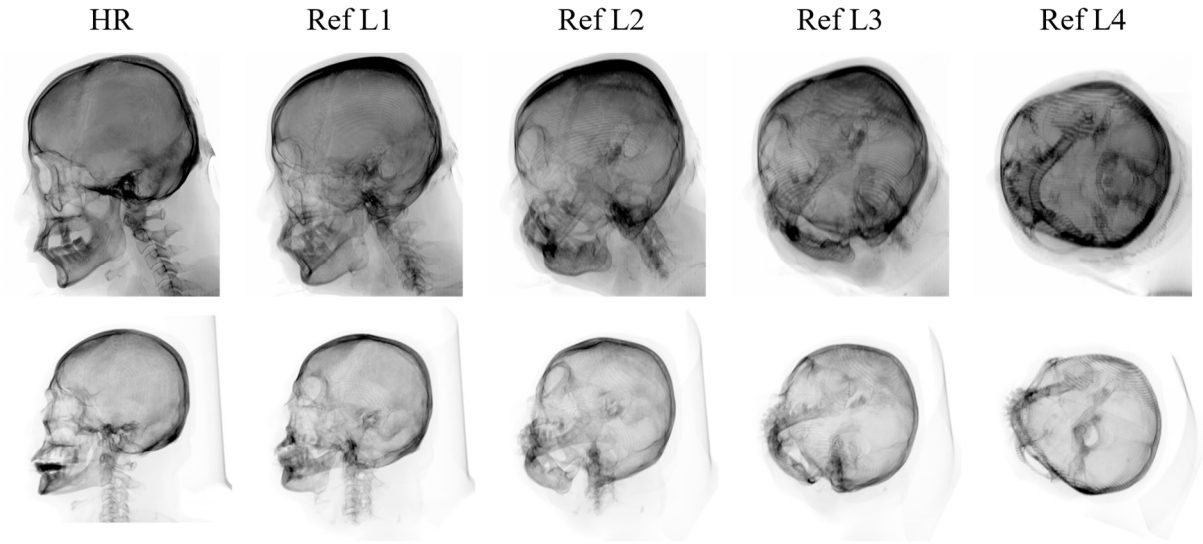

Figure 6-7: Examples from our synthetic X-ray testing set. From left to right are the HR image and the corresponding Ref images that the projection angle increment gradually.

Our synthetic X-ray image dataset can be regarded as containing five albums correspond to five patients. It is a relatively simple dataset compared to 1,883 albums in CUFED5. We did not apply feature matching as all synthetic X-rays images share similar structures, and we can control the projection angle of each image. We show an overview of our RefSR datasets in Table 6. In Section 6.3.2, our results indicate that the model can learn well on the smaller dataset, which indicates that the dataset size is sufficient for our model.

Table 6: Synthetic X-ray image RefSR dataset overview

Dataset	Patient number	Number of image pairs	Number of CT series used	Image Resolution
SynXray_S	1, 2, 3, 5	410	6	512 x 512
SynXray_L	1, 2, 3, 5	1350	6	512 x 512
Test	1, 2, 3, 4, 5	30	7	512 x 512

6.2.2 Chest X-ray Image Dataset for RefSR

We also build a real chest X-ray image RefSR dataset (ChestXrayRef) based on the National Institutes of Health (NIH) chest dataset: ChestX-ray8 [97]. The ChestX-ray8 dataset was collected from 1992 to 2015, consisting of 108,948 frontal-view X-ray images of 32,717 patients. It also has annotated disease labels, but only the image data is used in our study.

This dataset is built to validate further that our proposed method can work for broader usage in medical imaging. We also construct the dataset using a similar method in [57]. However, the ChestX-ray8 dataset only has X-ray images in a single frontal view. This makes it impossible to construct the dataset like synthetic X-ray images where the reference image is from a different angle of view. We consider the structure similarity of frontal view chest X-ray images and simply pick another chest X-ray image as the reference image. This certainly weakens the ability of RefSR method to reconstruct high-resolution textures. However, it has been proven that the reference image doesn't have to be closely related to the input image [57].

We randomly pick 2734 X-ray images from the ChestX-ray8 dataset in total for our RefSR dataset. All images are resized to 512 x 512 resolution from the origin 1024 x 1024 resolution. The training data has 1298 input images and 1298 reference images. After being randomly cropping into five patches, the training set has 6490 image patch pairs. The training set size of ChestXrayRef is equivalent to the SynXray_L dataset. 138 images are randomly chosen for the testing set and grouped into 23 image groups. We keep the same testing set structure, but the reference images no longer have a meaningful similarity level. Two example image groups in the testing set are shown in Figure 6-8.

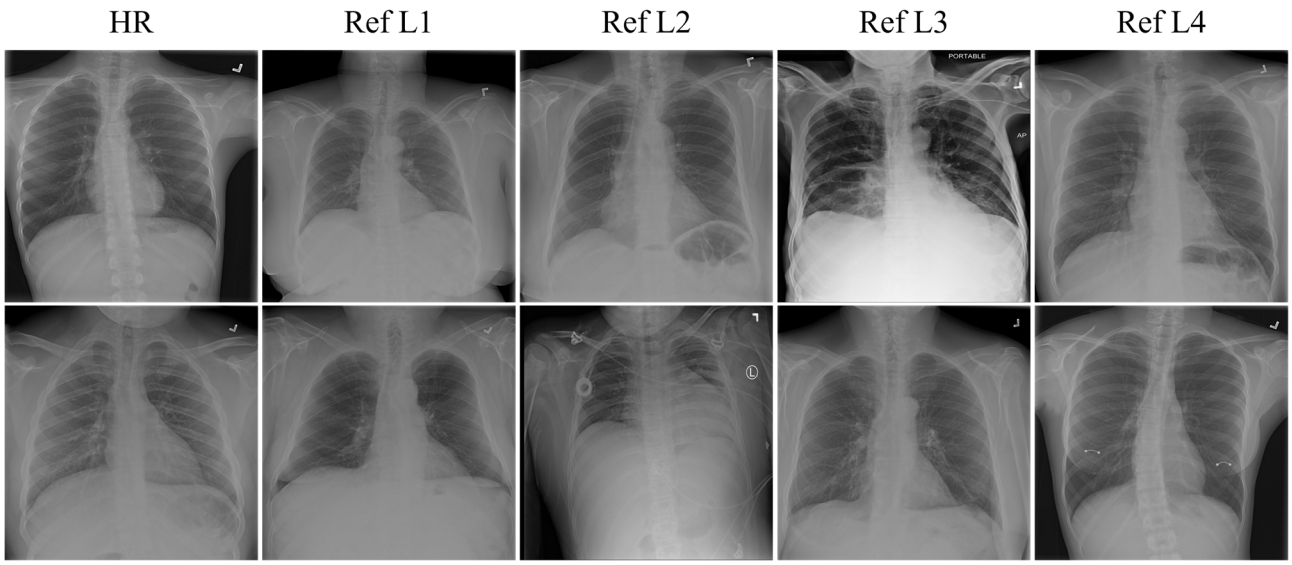

Figure 6-8: Examples from real Chest X-ray RefSR dataset. From left to right are the HR image (first column) and reference images.

6.2.3 SISR Dataset

We also build the corresponding Single Image Super-Resolution (SISR) dataset for synthetic head X-ray images to train conventional SISR methods for comparison. The concept of SISR is introduced in Section 3.2.1. The images we used are the SynXray_S dataset input images. The construction of a SISR dataset is straightforward. We simply use the input HR images to produce the corresponding LR images and form image pairs. We use the bicubic downsampling method from the PIL package to create the LR images. The synthetic SISR paired dataset has 410 image pairs. All the images in the SISR dataset are resized and have the same resolution as in the RefSR dataset.

6.3 Frequency Domain Loss for Super-Resolution

6.3.1 Training Details

We use a similar training parameter setting as in TTSR [20]. Pytorch 1.81 is used to implement our network. We train 100 and 50 epochs and pick the highest performance model for SynXray_S and SynXray_L dataset separately. The real X-ray image dataset ChestXrayRef is also trained for 100 epochs, and we pick the highest performance model. We use a batch size of 4. The learning rate is 1e-4. The parameters for Adam optimizer are set with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e - 8$. The

weight coefficient of the frequency domain loss is 1e-2. The weight coefficient for $\mathcal{L}_{rec}$, $\mathcal{L}_{adv}$, $\mathcal{L}_{per}$ are 1, 1e-3 and 1e-2, respectively. We train our network for a scaling factor of x4. We use bicubic kernel for rescaling the input and reference images. The model is trained with a Tesla P100 GPU. The training time for large and small synthetic datasets is around 10 and 4 hours, respectively. The training time for the real X-ray image dataset is around 18 hours.

6.3.2 Results with Synthetic X-ray Image Dataset

We compare our TTSR-FD as introduced in Section 5.2 with state-of-the-art methods, RCAN[72], ESRGAN[74] and TTSR[20]. TTSR as our baseline model has been modified for training grayscale images. Specifically, we duplicate the image grayscale channel into three RGB channels for natural images and then feed to the neural network. We retrain the RCAN and ESRGAN models with our dataset for comparison purposes as well. As mentioned in the previous section, we transfer our RefSR dataset into a SISR dataset for training RCAN and ESRGAN.

We evaluate our method on PSNR and SSIM, which are introduced in Section 2.11. The results are shown in Table 7. TTSR-FD achieves superior performance on both metrics. We also observe that TTSR-FD is strong on learning from a small dataset, indicating it can learn more from the data. This can be beneficial for real-life applications since medical data are hard to obtain. The best results are achieved at 60 epochs for TTSR and 70 epochs for TTSR-FD on the SynXray_S dataset. We get the best performance at 10 epochs for TTSR and 20 epochs for TTSR-FD on the SynXray_L dataset, respectively.

Table 7: PSNR/SSIM comparison among different SR methods on SynXray_S and SynXray_L datasets. The best results are in bold.

Method	SynXray_S	SynXray_L
RCAN	38.597/0.9482	38.880/0.9496
ESRGAN	34.483/0.9023	35.270/0.9140
TTSR	37.953/0.9393	38.115/0.9431
TTSR-FD (ours)	**39.009/0.9521**	**39.261/ 0.9514**

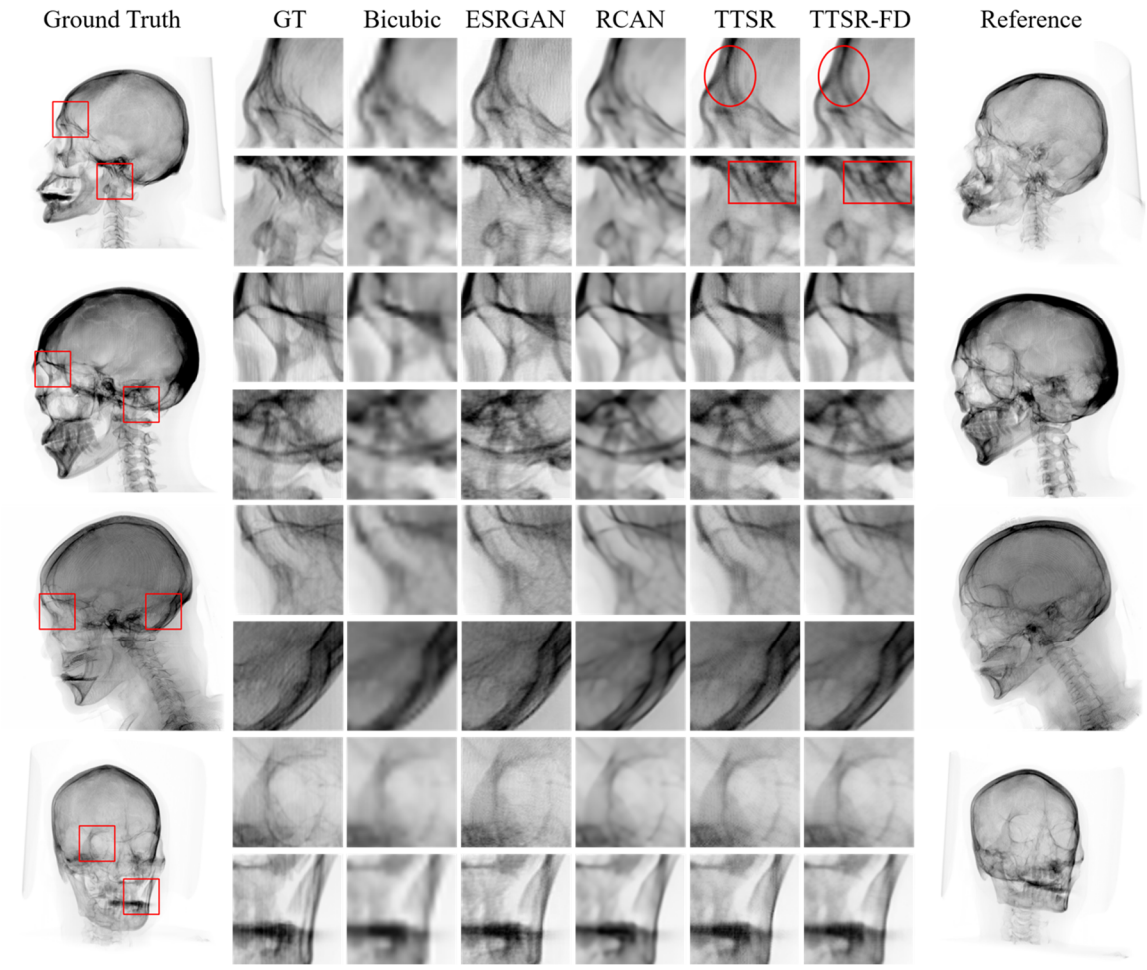

Figure 6-9: Visual comparison for our method. Testing images are generated from SAG, SAG, SAG and COR view of CT data, respectively. The ground truth (GT) image is the high-resolution version of the testing input image. The reference image (Ref) is also a high-resolution image where we use to transfer the high-resolution textures to the low-resolution testing input image. Both GT and Ref images are provided during testing. TTSR-FD is ours.

We show the visual comparison of our results on the testing set using our method trained with SynXray_S in Figure 6-9, considering that the visual difference between large and small datasets is minimal. RCAN shows the second-highest performance in the quantitative measurement, but we observe that the image is overly smoothed and has limited high-frequency details. ESRGAN gets the lowest PSNR and SSIM performance but shows visually appealing results. The main reason for such results is due to the method's focus on perceptual quality. We can observe artifacts, noise, and some unseen patterns created to satisfy human perception. This can be problematic for medical imaging. TTSR generates fine details with sharp edges but also generates artifacts in the resulting image. We can observe unseen patterns from the red circle area, and artifacts like black dots from the rectangle

area in TTSR results have been removed in TTSR-FD. Our method can generate fine details without generating much noise and artifacts.

6.3.2.1 Ablation Study

In this section, we further verify the effectiveness of our proposed method. We want to see if it's possible to achieve similar results by only increasing the weight coefficients of the reconstruction loss, without the use of frequency domain loss. For better understanding, we write the overall loss functions equation as follows:

$$\mathcal{L}_{overall} = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{fd}\mathcal{L}_{fd} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{per}\mathcal{L}_{per} \quad (6\text{-}1)$$

We set up TTSR (λ_{rec}= 1.01) to match the same ratio of weight coefficients distribution as TTSR-FD. The weight coefficients of reconstruction loss $\mathcal{L}_{rec}$ is increased from 1 to 1.01. We then compare it with our TTSR-FD model. The weight coefficients of TTSR-FD for $\mathcal{L}_{rec}$ and $\mathcal{L}_{fd}$ are 1 and 0.01. The weight coefficients of other loss function $\mathcal{L}_{adv}$, $\mathcal{L}_{per}$ are the same for both models. Our results are shown in Table 8.

Table 8: Ablation study for frequency domain loss with SynXray_S dataset

Method	PSNR	SSIM
TTSR (λ_{rec}= 1)	37.953	0.9393
TTSR (λ_{rec}= 1.01)	38.245	0.9405
TTSR-FD(λ_{rec}= 1, λ_{fd}= 0.01)	**39.009**	**0.9521**

We can see that compared to only match the weight coefficient of the reconstruction loss to TTSR-FD, the proposed frequency domain loss significantly improved the model performance. We also transfer our testing results into magnitude spectrums to see the different patterns in the frequency domain, as shown in Figure 6-10. We can find out that only increasing the weight coefficient of the reconstruction loss could not improve the similarity of high-frequency patterns in the frequency domain. As shown in red arrows, we observe that TTSR (λ_{rec}= 1.01) does not learn the vertical line patterns.

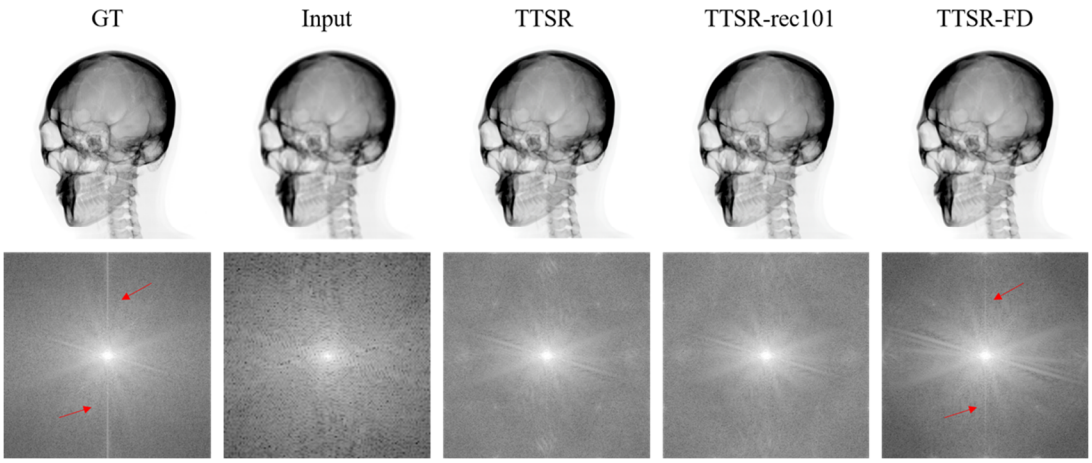

Figure 6-10: Ablation study on frequency domain loss. Vertical line patterns indicated in red arrows in the ground truth (GT) image are only learned by TTSR-FD.

We study a broader range of reconstruction loss weight coefficients to validate if an even higher reconstruction loss weight coefficient could boost the method's performance. We test the λ_{rec} from 0.01 to 0.5. We first increase the weight coefficient with an increment of 0.01 until 0.09, then we use an increment step of 0.1 until 1.5. We show the results with bar and line graphs of PSNR and SSIM, as shown in Figure 6-11.

We observe no relation between the model performance and the weight coefficient λ_{rec} when increasing between 0.01 and 0.09. And there is no consistent boost of performance when increasing the weight coefficient to 0.5. We can observe that the weight coefficient of the reconstruction loss does not have a certain impact on the model performance. The performance of the TTSR model could not reach the TTSR-FD model performance by increasing the weight coefficient of the reconstruction loss.

We also study how the weight coefficient of FD loss affects the model performance. We show the model quantitative results using the FD loss weight coefficient range from 0.1 to 0.0001 in Table 9. We can observe that our chosen weight coefficient value of 0.01 has the best performance.

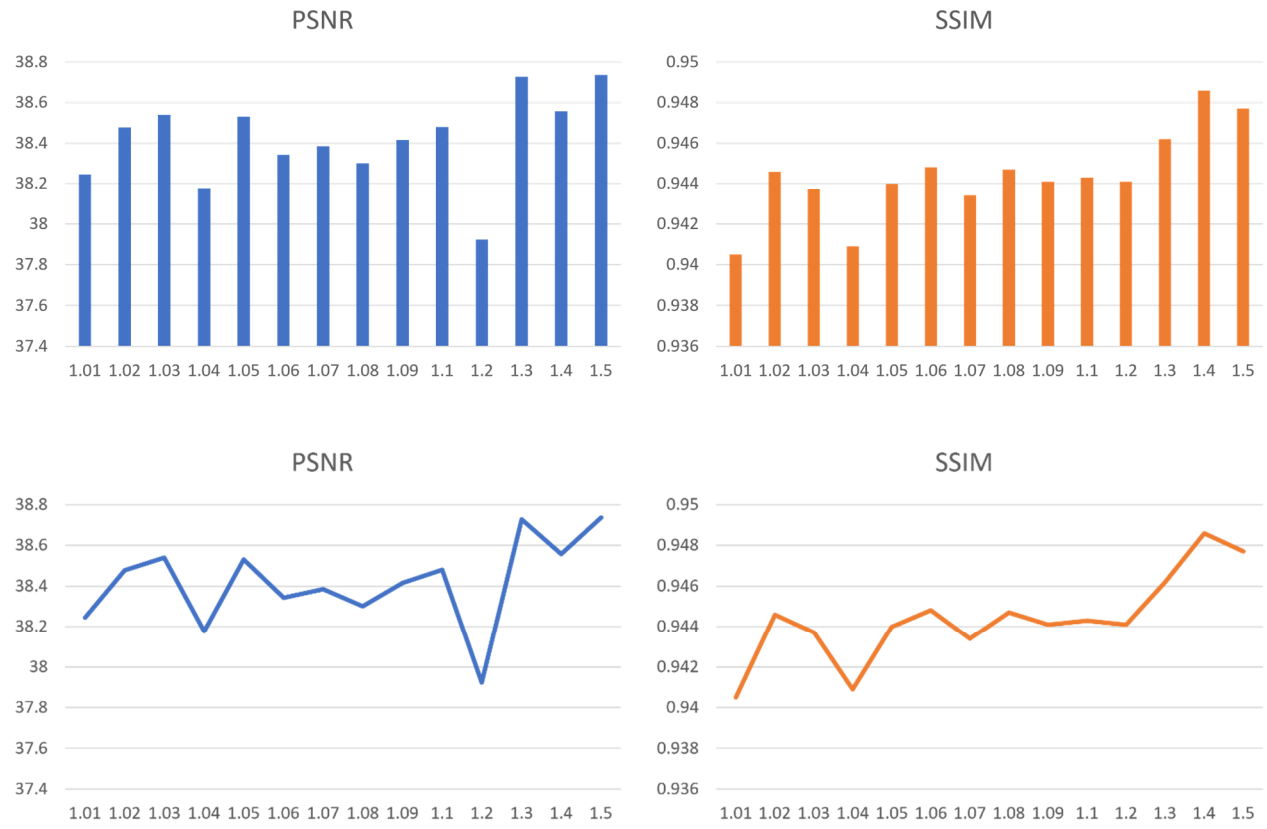

Figure 6-11: Graphs of extended ablation study on reconstruction loss weight coefficient. PSNR and SSIM are displayed in bar and line graphs.

Table 9: The weight coefficient of frequency domain loss λ_{fd} effect on TTSR-FD model performance.

$\boldsymbol{\lambda_{fd}}$	0.1	0.01	0.001	0.0001
PSNR/SSIM	38.194/0.9420	**39.009/0.9521**	37.668/0.9404	38.001/0.9423

6.3.3 Results with Real Chest X-ray Image Dataset

To prove that our method can work for more medical imaging applications, we also evaluate our proposed TTSR-FD on our real X-ray image dataset ChestXrayRef. We retrain TTSR and TTSR-FD models for real Chest X-ray image dataset. Table 10 shows the testing results in the quantitative evaluation of PSNR and SSIM. We found that the TTSR and TTSR-FD models gained similar quantitative performance. TTSR has a slightly higher PSNR score than TTSR-FD, and TTSR-FD got a higher SSIM score compared to TTSR. We do not see a performance boost on quantitative evaluation adopting frequency domain loss on the real Chest X-ray images dataset.

Table 10: PSNR/SSIM comparison for ChestXrayRef dataset

Method	PSNR	SSIM
TTSR	**36.767**	0.9457
TTSR-FD	36.766	**0.9470**

However, we find out from visual comparison that TTSR-FD generates significantly fewer artifacts compared to TTSR results. We show the visual comparison in Figure 6-12. We already observe that frequency domain loss can remove artifacts and noise for the synthetic X-ray image dataset. The artifacts and noise in the real chest X-ray image dataset are more evident than those in the synthetic dataset. The artifacts are even more perceivable when contrast is higher. Our TTSR-FD can still remove such artifacts in the TTSR results for the real X-ray image dataset. It did not get any superior results on the quantitative comparison. But we observe image quality improvement with TTSR-FD in general. The perceptual quality of the resulting super-resolution chest X-ray images is improved by frequency domain loss.

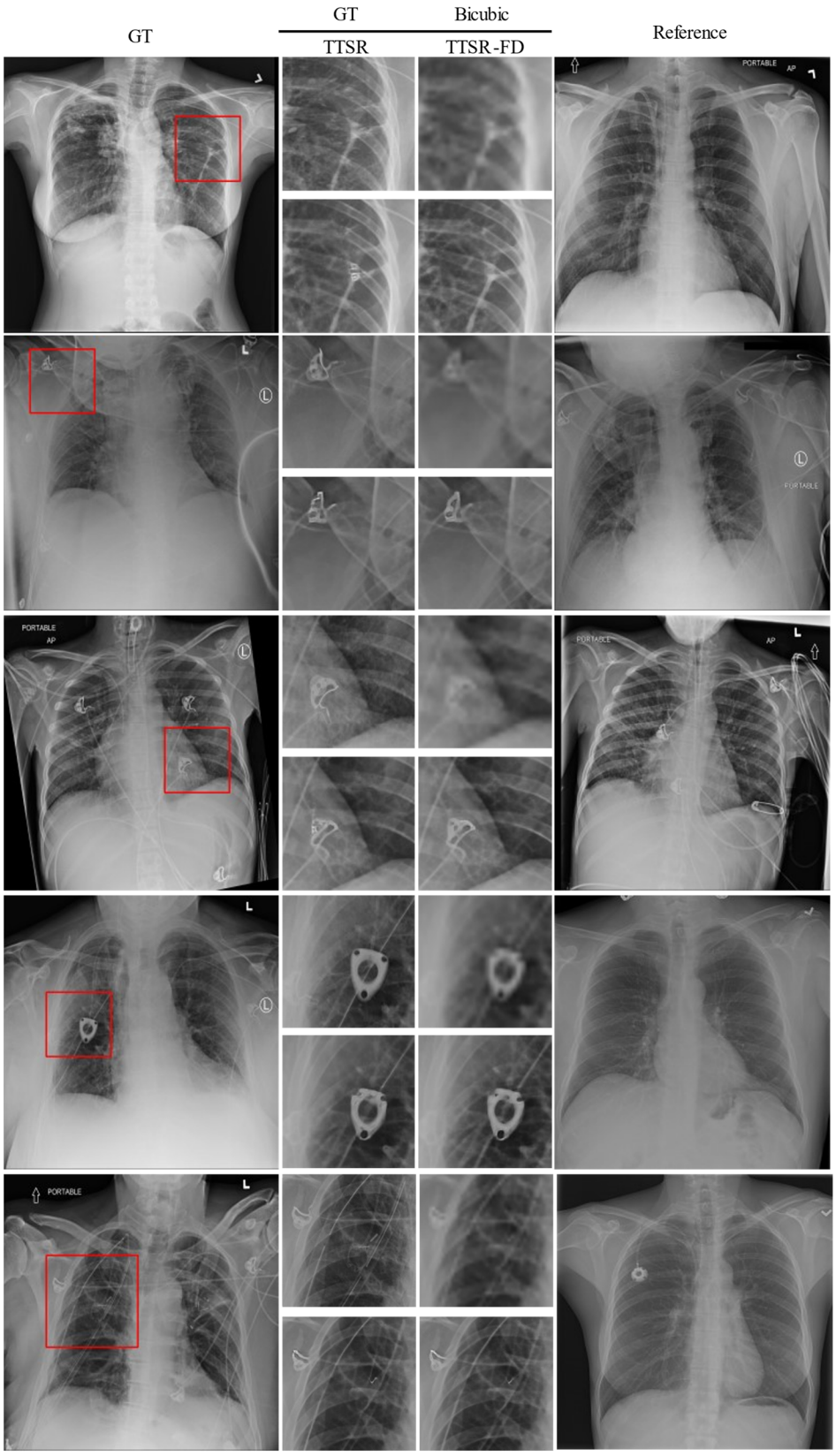

Figure 6-12: Visual comparison for our results on the ChestXrayRef testing set.

6.4 Adaptation Network for Super-Resolution

6.4.1 Training Details

We use a similar training parameter setting as in TTSR [20] to train our TTSR-DMK model. We use Pytorch 1.81 to implement our network. We train 100 epochs and pick the highest performance model for the SynXray_S dataset. The dataset is processed differently for the adaptation network compared to how we train the TTSR-FD model, where three downsampling kernels are randomly used during data preprocessing. Specifically, we randomly use PIL Bicubic, PIL Bilinear, and PIL Nearest to downsample HR images to LR images when loading our dataset. We use a batch size of 4. The learning rate is 1e-4, and Adam optimizer parameters are $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e - 8$. The weight coefficient for $\mathcal{L}_{rec}$, $\mathcal{L}_{adv}$, $\mathcal{L}_{per}$ are 1, 1e-3 and 1e-2, respectively. The weight coefficient of the dual regression loss is 1e-3. We build our network on an upsampling factor of four. We train our model with a Tesla P100 GPU. The adaptation network training time is around 7 hours. We observe an increase in training time for 3 hours compare to the baseline model TTSR.

6.4.2 Results

To evaluate the effectiveness of our TTSR-DMK that is introduced in Section 5.3 to solve the adaptation problem, we train a TTSR and TTSRD (TTSR-Dual) model with only the PIL Bicubic kernel used as downsampling kernel. And we train a TTSR-MK model with three downsampling kernels in the same way we train our TTSR-DMK. After training these models, we then evaluate our results on multiple downsampling kernels. This includes kernels used during training (PIL Bicubic, PIL Nearest, PIL Bilinear) and kernels not used during training (OpenCV Bicubic, OpenCV Linear).

6.4.2.1 Results on kernels used during training

We evaluate our results with each of the downsampling methods we used in the training period. Table 11 shows the quantitative evaluation results on these three kernels. TTSR achieves promising results on PIL Bicubic, which is used during its training. We observe a significant drop in performance on the other two kernels with TTSR due to adaptation issues. TTSRD gains better performance than TTSR in general, proving that the closed-loop method can improve performance for conventional RefSR training. Our TTSR-DMK shows the best performance for all three kernels

in general. We pick testing images produced with PIL Nearest kernel to compare our results in Figure 6-13 visually. TTSR shows lots of artifacts and noise during testing due to adaptation issues. TTSR-MK achieves better image quality by training with multiple kernels. TTSR-DMK has the best quality and can generate more details as marked in red circles.

Table 11: PSNR/SSIM performance of super-resolution models on images produced by different degradation methods used in adaptation network training

Method	PIL Bicubic	PIL Nearest	PIL Bilinear
TTSR	38.286/0.9429	31.191/0.8779	37.794/0.9418
TTSRD	38.486/0.9432	31.292/0.8795	37.817/0.9424
TTSR-MK	38.004/0.9426	35.467/0.9256	38.386/0.9417
TTSR-DMK	**38.554/0.9435**	**35.714/0.9301**	**38.454/0.9440**

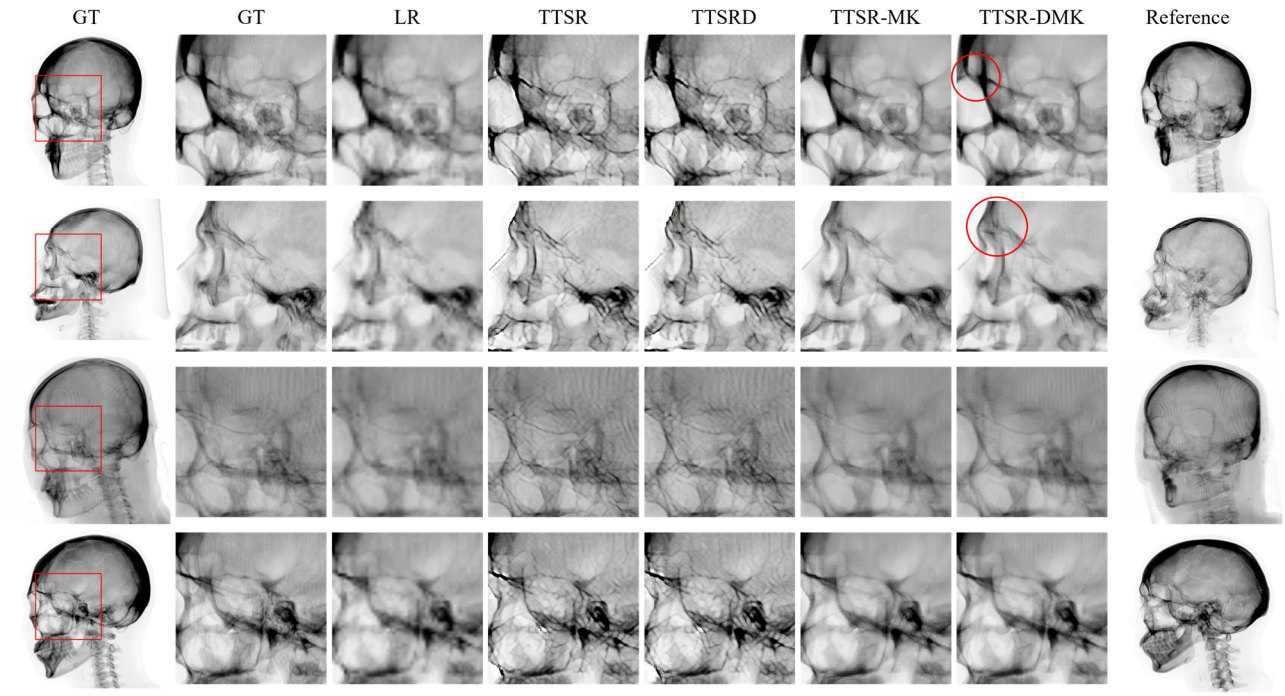

Figure 6-13: Visual comparison of our method on testing with PIL Nearest downsampling Kernel.

6.4.2.2 Results on kernels not used during training

We evaluate our method on two OpenCV kernels that are not used during training. We pick OpenCV Bicubic and OpenCV Bilinear kernels for our evaluation. They generate very different LR

images compared to the PIL kernels. The TTSR-DMK method gets the best results for both degradation methods, as shown in Table 12. We can observe that the TTSR model gains significantly lower performance than the previous section due to the adaptation issue. We show a visual comparison with TTSR and TTSR-DMK on results tested with images produced with OpenCV Bicubic kernel in Figure 6-14. The TTSR-DMK resulting image quality is lower than what we got in the previous section's experiments with kernels used in training. However, it still has a significant improvement in removing artifacts and noise caused by adaptation issues.

Table 12: PSNR/SSIM performance of super-resolution models on images produced by degradation methods that not used in adaptation network training

Method	OpenCV Bicubic	OpenCV Linear
TTSR	31.381/0.8821	32.863/0.9010
TTSRD	31.609/0.8839	33.210/0.9038
TTSR-MK	34.181/0.9091	34.051/0.9071
TTSR-DMK	**34.285/0.9132**	**34.185/0.9141**

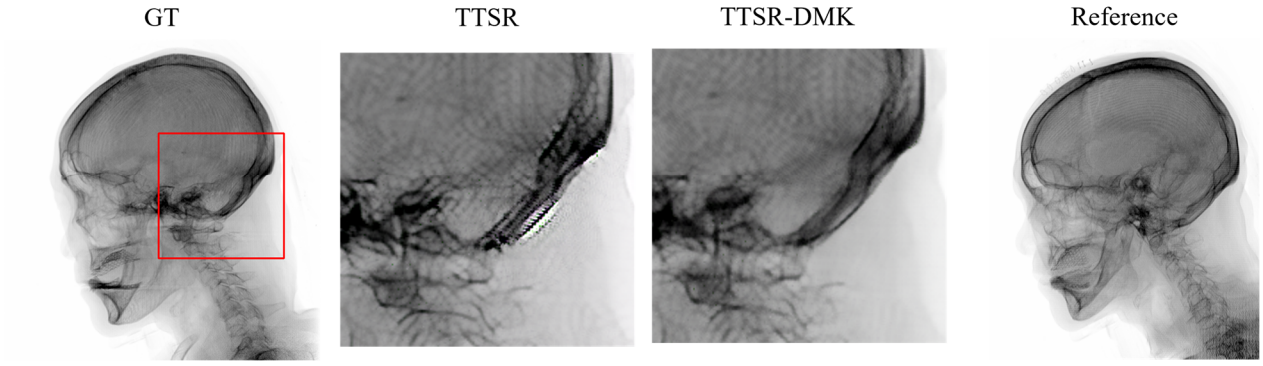

Figure 6-14: Visual comparison of TTSR and TTSR-DMK results when testing images produced by OpenCV Bicubic downsampling kernel.

6.5 Discussion

Our synthetic X-ray image generation method produces high-quality images in general. It can be positioned in a place between the traditional ray-tracing method [10], [67], [98] and recent deep learning-based methods [6], [12]. Conventional methods often aim for 2D/3D registration that

focuses on position alignment. As a result, these methods are not realistic and lack good contrast on soft tissues or organs. Deep learning-based methods can produce realistic images but with extensive training data and in a complex matter. They aim to generate highly realistic images that can even replace real X-ray images as deep learning training data. However, the images produced by GAN-based deep learning methods such as image-to-image translation or style transfer have certain issues, as shown in Figure 6-15. Because GAN could produce artifacts and create texture details to look realistic but not anatomically correct. Benefit from the dedicated dynamic and multi-segment lookup tables, our method can generate much higher quality images than the conventional method without the need of refining by a deep learning model. It is not as realistic as results produced by modern deep learning methods, but it offers better precision in anatomical structure. The correctness of the anatomical structure of the synthetic X-ray images is important, especially when such images are used as reference images during real surgical operations or used for VR training simulations.

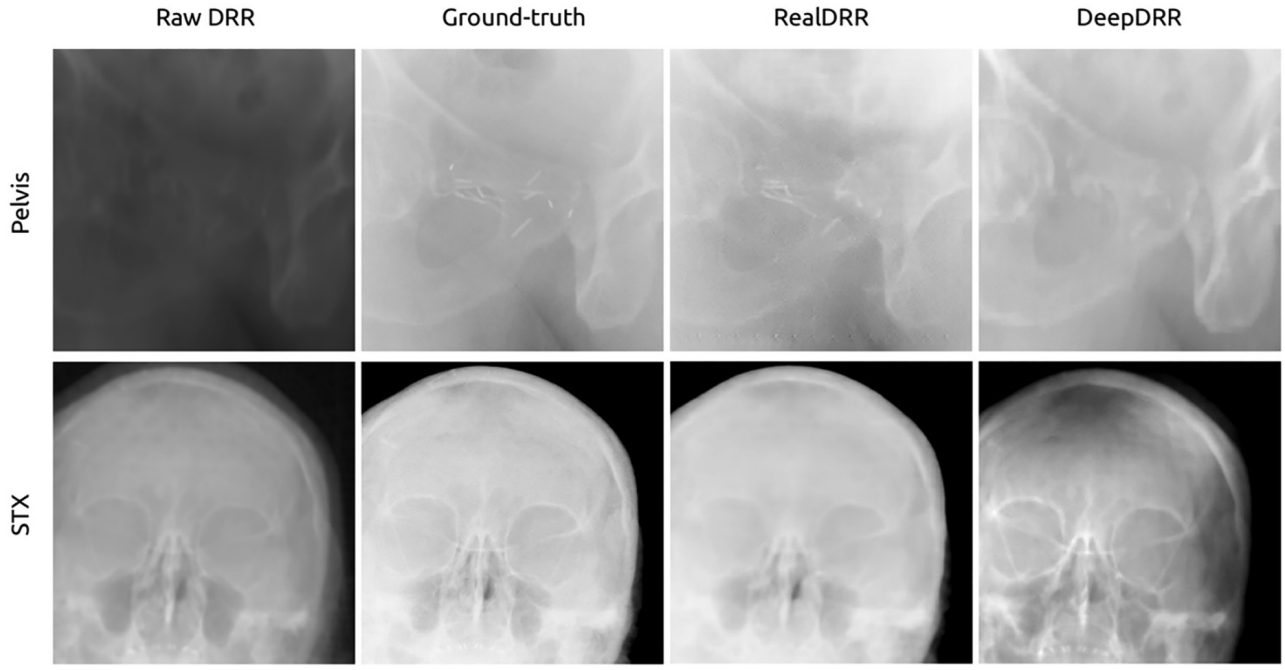

Figure 6-15: Sample results using deep learning methods RealDRR [12] and DeepDRR [6]. The resulting images are realistic, but we can notice that deep learning methods introduce anatomical structures that are not present in Ground-truth. Some structures are missing in the deep learning methods results as well. Figures are reprinted from [12] with auto permission.

On the other hand, the method works well under most scenarios but produces lower-quality images in certain angles of view. This issue can be mitigated by introducing more data in the

generation method, i.e., including more than one view of CT data. The generation process is always conducted with single view CT data in our experiments, such as the sagittal or coronal views. Our generation method can utilize multiple view CT data with an optimization algorithm build based on projection angles. This could mitigate the image quality drop shown in Figure 6-2. A CT scan can be digitally processed into multiple views, as long as the need for them is known during the scan. Our CT data source is not dedicated to synthetic X-ray image generation. Thus some series only have one or two views.

We evaluate our proposed frequency domain loss in Sections 6.3.2 and 6.3.3. The frequency domain loss works well for synthetic X-ray images on improving PSNR and SSIM scores but not for real X-ray images. In general, the synthetic X-ray image dataset has higher PSNR and SSIM scores compare to the real X-ray image dataset. Frequency domain loss works well on removing artifacts and noise for both datasets, especially for real X-ray images. The differences in the results on the two datasets could be due to the dataset characteristic. The synthetic X-ray image dataset and real X-ray image dataset are different in several ways. Firstly, the synthetic dataset is human head data, while the real X-ray image dataset is human chest data. Chest X-ray images can be fundamentally more complicated than head X-ray images and more difficult for the model to learn. Secondly, real X-ray data has the X-ray characteristic that synthetic data does not possess. As mentioned in Section 3.1, our generation method does not simulate beam hardening and scattering in real X-ray images. In other words, a synthetic X-ray image is more of a pure image. Thus, it is more difficult when reconstructing real X-ray images. This could also cause the model to generate more artifacts for real X-ray images due to noise and other characteristics that real X-ray images have. Thirdly, the real X-ray image RefSR dataset is not constructed the same way as the synthetic one. The synthetic X-ray image RefSR dataset is built in an ideal situation where we can generate reference images with any projection angle we prefer. This is beneficial during training. The real X-ray RefSR dataset can only be built under the worst-case scenario, where the reference image is not very related to the LR input image. In general, we proved our frequency domain loss is a powerful tool to improve image quality, especially for artifacts removal.

We propose TTSR-DMK that achieves good results in solving the adaptation problem. It does have certain limitations, such as require more data to gain a good result. This is because by training the data with different kernels, we are essentially reducing the amount of data for a single kernel. We also find out it could not run together with our TTSR-FD model. Applying these two techniques

together causes the performance to drop. We analyze the problem and find out that the model is unstable when both techniques are used. One possible reason is that the overall loss function being too complicated and made the model hard to converge. Another possible assumption is that the frequency domain loss focuses on details of the textures, while we want a trade-off between details and generalization in the adaptation network. Thus, the two approaches are contradicting between each other.

Combining our DRRs and deep learning super-resolution method, we can efficiently produce high-resolution X-ray images from CT data even with large gap slices. Deep learning has been used to improve synthetic X-ray image quality, as mentioned in previous paragraphs. These methods usually use a conventional fast generation method to produce lower-quality X-ray images and then use neural networks to enhance the image quality. We also use deep learning to improve the image quality but focus on increasing the resolution with fewer artifacts instead of style transfer. Deep learning for style transfer tends to produce plausible results without anatomical precision. Our DRR method can produce finner quality X-ray images compare to conventional methods with a scarify on processing speed. Then we use super-resolution methods to increase the resolution of images.

Chapter 7. Synthetic X-ray VR Application

7.1 Overview

In this section, we briefly introduce an application based on our synthetic X-ray image generation algorithm. As we mentioned in Sections 1.1 and 2.2, synthetic X-ray images can be used in virtual reality (VR) simulation to train physicians in fluoroscopy-guided intervention procedures. We implement our method in Unity3D with C# to create a VR training simulation for C-arm-based image guiding intervention procedures. The VR simulation we discuss here is a demo version. In the project, our main role is to develop the synthetic X-ray image generation algorithm for Unity3D. Dr. Jae Chul Koh from Korea University Anam Hospital conducts interaction and graphic development.

7.2 Implementation

The CT data used for testing in this demo is from patient 2 and patient 4. Image interpolation is performed to gain better generation results. The interpolation only needs to be performed once as a data pre-processing step. For efficiency consideration, our generation method produces 256 x 256 resolution synthetic X-ray images. We choose to pre-generate synthetic X-ray images to achieve real-time performance.

In our C# implementation, we only use Unity3D API to implement our algorithm without third-party libraries. We replace most python functions with their counterpart in Unity3D and build the rest of the functions in C#. The Vector3 API serves as the numpy array function and makes it convenient to perform any transformation in Unity3D. It also runs much faster compared to the implementation in Python (without Numba accelerating). Quaternion handles all the transformation calculations efficiently. Texture2D works as image storages to transfer our calculation results to objects in Unity3D. To modify the pixel information in Texture2D, we use RenderTexure to rewrite the texture information. Color API is used to store the pixel value calculation results. Color objects are then used to update the Texture2D object. We list the majority of Unity API we used when developing the synthetic X-ray image generation method in Unity3D, as shown in Table 13.

Table 13: Unity API usage in our work

Unity API	API Usage
Texture2D	Texture2D is used to store and display images in Unity3D
Vector3	Vector3 is used to process and store arrays in Unity3D
Color	Color is used to process and store the calculated pixel value. Its value range from 0 to 1.
Quaternion	Quaternion is used to perform efficient rotation transformation.
Color32	Different from Color, Color 32 represent RGBA colors in 32-bit format with pixel values range from 0 to 255.
Color32.Lerp	Color.Lerp is used to perform interpolation in Unity3D.
RenderTexture	RenderTexture is used to make the texture readable for calculations.

We create a similar algorithm to the X-ray generation algorithm for projecting the 3D needle. The dimension of the needle is modelled in Unity3D. The needle and the synthetic X-ray image are two objects in the simulation scene and overlapped for display. The needle is tracked in real-time. A separate transform and projection algorithm based on our synthetic X-ray image algorithm is built for the needle.

A general process of our synthetic X-ray image calculation based on our method in Figure 4-1 for Unity3D can be described as follows:

- Initiate a new Texture2D object.
- Load interpolated CT data into $coordinates$ and $pixels$, where $coordinates$ store in Vector3 and $pixels$ in a float list.
- Use Quaternion to perform transformations on $coordinates$.
- Register $coordinates$ and $pixels$ into a dictionary and calculate the pixels value for the synthetic X-ray image.
- Store the calculated values in Color objects.
- Update Color to the Texture2D object.

- Display the Texture2D in Unity3D or save it in the local machine for pre-generation.

The synthetic X-ray images are pre-generated in the demo because of the expensive computational cost. In the demo, we generate synthetic X-ray images in projection angles ranging from -120º to 120º on the x-axis and -30º to 30º on the y-axis with a gap of 5º, which covers the major usage scenarios of the C-arm machine. The C-arm machine is set to rotate with a gap of 5º.

7.3 Results and Discussion

We apply our synthetic X-ray image generation method to build a VR training simulation preview demo in Unity3D. The simulator can track the needle in real-time and project the needle to the pre-generated synthetic X-ray images. The simplified scene contains the C-arm machine, bed, head bone, and three raw images as shown in Figure 7-1. The left raw images consist of two overlapping images. One image displays the synthetic X-ray image, and the other displays the tracked needle. The right raw image shows the transform parameter when connected to a VR device.

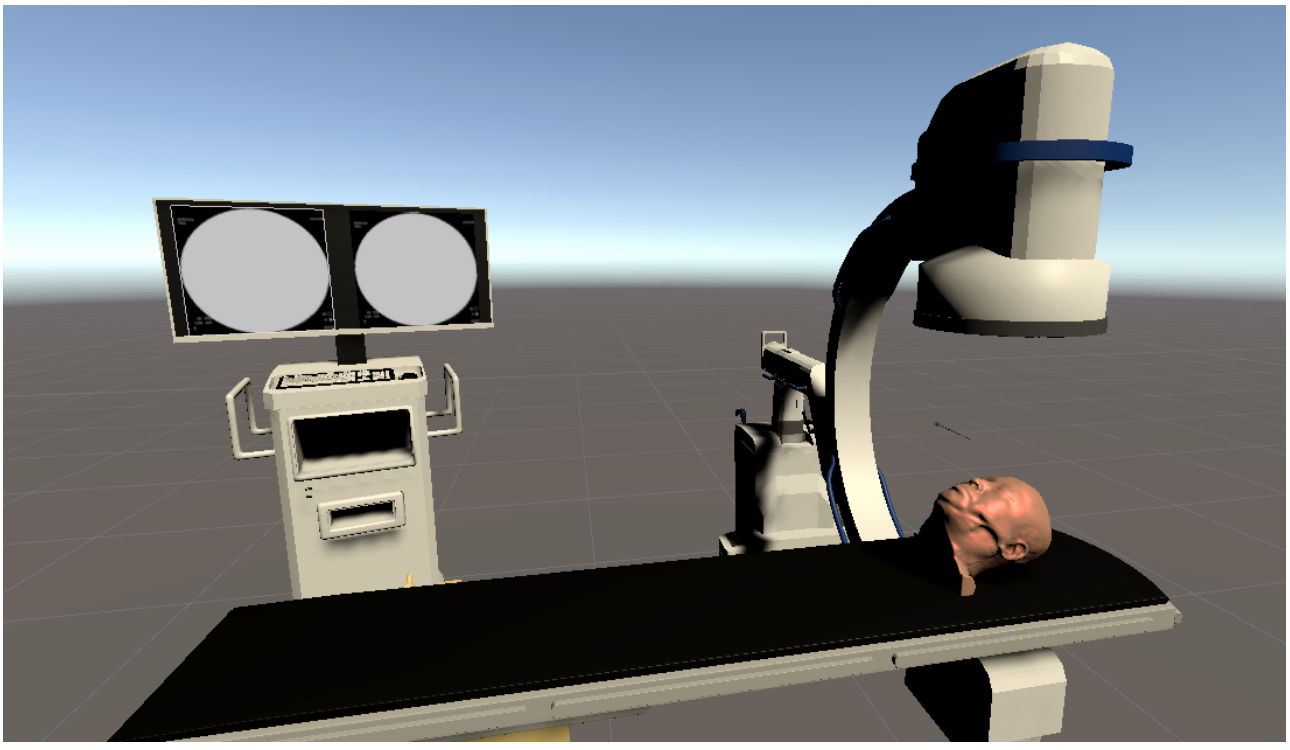

Figure 7-1: The Operating scene for the VR training simulation demo.

A running testing scene performed on PC is shown in Figure 7-2. The coordinates' monitor only works with the VR environment and is in its default number. During testing, we can rotate the

C-arm machine and operate the needle. The monitor update each new synthetic X-ray image from pre-generated data correspond to the C-arm rotation angle in real-time. The needle shows up in the image when it enters the border area set around the head bone and within the projection area. Our proposed synthetic X-ray image generation method provides quality synthetic X-ray images with fine details and contrast. The needle movement shows good accuracy and can be tracked in real-time and displayed on top of the synthetic X-ray images.

Our generation method used in this application enables high-quality synthetic X-ray image display, which is pointed out in [27] as a performance bottleneck of VR C-arm simulation. The quality and resolution of images are important in VR applications as the resolution of the VR simulation scene is high. In the future, this application could be further improved by integrating the super-resolution method. This could reduce the generation time by generating lower resolution images and also make a zooming-in function applicable.

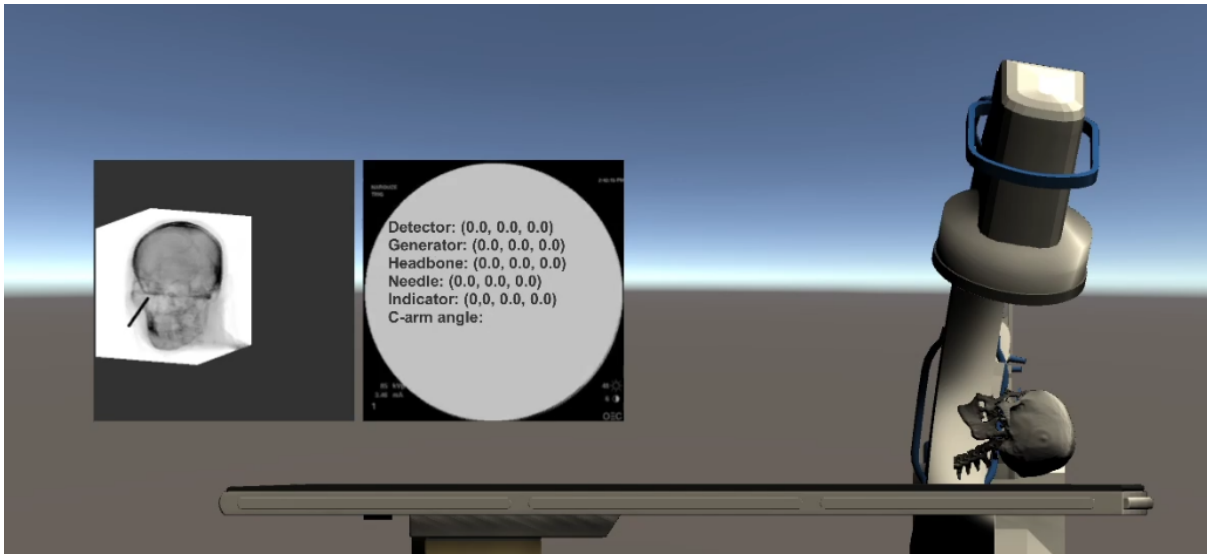

Figure 7-2: A testing scene of our VR training simulation demo.

The demo also has certain limitations. Firstly, we use pre-generated data, and it can work for training purposes but reduce the realistic aspect of the simulator. It is also not feasible to pre-generate all the angles of synthetic X-ray images, considering that it would need near 130,000 images for each set of CT data. This is also the reason we develop a super-resolution method that aims to up sample low-resolution images, which takes a much shorter time to generate. Secondly, the needle tracking algorithm can also be improved if we generate synthetic X-ray images in real-time. The current synthetic X-ray image and the needle are displayed in two images while the needle has a transparent

background. This does not affect the accuracy theoretically, but it could improve algorithm efficiency if we combine the codes for the synthetic X-ray image and the needle together. Thirdly, the simulator does not replicate all the characteristics of a real C-arm machine. It needs further development to have more features such as ray settings, more freedom of rotation, etc.

Chapter 8. Conclusion

8.1 Summary of Contributions

We aim to improve synthetic X-ray image generation from CT data in the generation and super-resolution method. In summary, we made the following contributions in this thesis.

Firstly, we proposed an alternate projection method with lookup tables to generate synthetic X-ray images from CT data. Our experiments on human head CT data shown improved image quality compared to conventional methods. The projected synthetic X-ray images demonstrated sufficient accuracy with rich content and good contrast in most projection angles. Our method could fit in between conventional and deep learning generation methods, providing decent quality images. It could generate arbitrary view synthetic X-ray images from raw CT images instead of DICOM files. The simplicity of the algorithm made our method easy to implement in other programming languages or platforms. Our method is also suitable for vast data generation for deep learning training data.

Secondly, we developed TTSR-FD, a reference-based super-resolution architecture for synthetic X-ray images. It utilized frequency domain loss to improve the resulting image quality. A synthetic X-ray image and real Chest X-ray image datasets for RefSR were also created for our experiments. We demonstrated that an additional loss function in the frequency domain could improve the image quality for synthetic X-ray image super-resolution. Our work enables the possibility of generating high-quality synthetic X-ray images in real-time for Image Guiding Systems and VR simulations. We also evaluated our method on a real chest X-ray image dataset, proving that our method could improve image quality and remove artifacts for a broader range of medical images.

Thirdly, we proposed TTSR-DMK to mitigate the adaptation issue that most deep learning super-resolution methods have. The adaptation issue is that most SR methods are trained with paired LR and HR images, where LR image is created from HR image with a single downsampling kernel. We used multiple degradation methods during training to generalize the model. We adopted a dual model to improve our model performance further. It was proven that the dual model could improve conventional and multi-kernel adaptation networks for RefSR. Our work can generalize the super-

resolution model to make it better for downsampling kernels not seen during training with a trade-off on increasing the training time.

Our innovation lies in combining good quality synthetic x-ray generation and deep learning-based super-resolution. We aim to increase the resolution even higher to keep the anatomical structure correctly, rather than using deep learning-based style transfer. The technical contribution in this work can be divided into three parts: synthetic X-ray image generation, synthetic X-ray image super-resolution, and adaptation network for synthetic X-ray image super-resolution.

i. We proposed to use an alternative way to generate synthetic X-ray images. We adopted a matrix-based projection method and a dynamic multi-segment lookup table with tissue radiographic opacity parameters.
ii. A frequency-domain loss is proposed to improve the performance of a RefSR method for synthetic X-ray images. We achieved state-of-the-art results on synthetic X-ray image super-resolution with our proposed TTSR-FD.
iii. We also contributed to the adaptation issue of super-resolution methods. To address this issue, we proposed to adopt multiple downsampling kernels to create LR images from HR images during the data pre-processing period and built a dual model as closed-loop to learn an inverse mapping from HR to LR images. Our TTSR-DMK gained improvements on both conventional and adaptation SR networks.

8.2 Future Work

There are also certain limitations related to this work. The synthetic X-ray image generation method still has room for improvements in computation time. It is possible to increase the algorithm efficiency by optimizing the data processing part. We tested on human head CT data, and the setting of the lookup table needs to be readjusted when CT is taken for other body parts. We produced synthetic X-ray images in orthogonal projection in our experiments, and we would like to extend to perspective projection in the future. We also want to improve the image quality when the projection angle is far away from the original angle of the CT data. A more dedicated interpolation method such as deep learning-based interpolation could also improve our resulting image quality. Moreover, it would be interesting to see if we can use a deep learning model to learn the optimal lookup table

setting for different CT data from various human body structures. It would certainly need the support of more available paired CT and X-ray data.

We want to explore more possible loss variants for the frequency domain loss for our proposed TTSR-FD method. We tested our method on synthetic and real X-ray images, and we hope to extend it on other medical images such as MRI or ultrasound images. We also would like to work on other application scenarios such as image denoising or inpainting with frequency domain loss in the future. It would also be interesting to see if the frequency domain loss can be helpful for natural image super-resolution.

The TTSR-DMK method works well for solving the adaptation problem. However, there is also room to improve since the performance gap between adaptation and conventional SR networks is still huge. We want to work more on how to train with more kernels efficiently. It is also worth exploring how to apply both frequency domain loss and dual regression loss in training the same neural network model.

References

[1] "On a New Kind of Rays," *Nature*, vol. 53, no. 1369, Art. no. 1369, Jan. 1896, doi: 10.1038/053274b0.

[2] D. C. Shelledy and J. I. Peters, *Respiratory Care: Patient Assessment and Care Plan Development*. Jones & Bartlett Publishers, 2014.

[3] W. Birkfellner, J. Wirth, W. Burgstaller, B. Baumann, H. Staedele, B. Hammer, N. C. Gellrich, A. L. Jacob, P. Regazzoni, and P. Messmer, "A faster method for 3D/2D medical image registration--a simulation study," *Phys Med Biol*, vol. 48, no. 16, pp. 2665–2679, Aug. 2003, doi: 10.1088/0031-9155/48/16/307.

[4] D. R. Allen, "Simulation Approaches to X-ray C-Arm-based Interventions," p. 85.

[5] P. Korzeniowski, R. J. White, and F. Bello, "VCSim3: a VR simulator for cardiovascular interventions," *Int J CARS*, vol. 13, no. 1, pp. 135–149, Jan. 2018, doi: 10.1007/s11548-017-1679-1.

[6] M. Unberath, J.-N. Zaech, S. C. Lee, B. Bier, J. Fotouhi, M. Armand, and N. Navab, "Deepdrr–a catalyst for machine learning in fluoroscopy-guided procedures," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2018, pp. 98–106.

[7] B. Bier, M. Unberath, J.-N. Zaech, J. Fotouhi, M. Armand, G. Osgood, N. Navab, and A. Maier, "X-ray-transform Invariant Anatomical Landmark Detection for Pelvic Trauma Surgery," Mar. 2018, Accessed: Apr. 11, 2021. [Online]. Available: https://arxiv.org/abs/1803.08608v1

[8] X. Ying, H. Guo, K. Ma, J. Wu, Z. Weng, and Y. Zheng, "X2CT-GAN: Reconstructing CT from Biplanar X-Rays with Generative Adversarial Networks," May 2019, Accessed: Apr. 11, 2021. [Online]. Available: https://arxiv.org/abs/1905.06902v1

[9] M. Touchette, R. Newell, C. Anglin, P. Guy, K. Lefaivre, M. Amlani, and A. Hodgson, "The effect of artificial X-rays on C-arm positioning performance in a simulated orthopaedic surgical setting," *International Journal of Computer Assisted Radiology and Surgery*, pp. 1–12, 2020.

[10] R. L. Siddon, "Fast calculation of the exact radiological path for a three-dimensional CT array," *Medical physics*, vol. 12, no. 2, pp. 252–255, 1985.

[11] H. Zhao and A. J. Reader, "Fast ray-tracing technique to calculate line integral paths in voxel arrays," in *2003 IEEE Nuclear Science Symposium. Conference Record (IEEE Cat. No.03CH37515)*, Oct. 2003, vol. 4, pp. 2808-2812 Vol.4. doi: 10.1109/NSSMIC.2003.1352469.

[12] J. Dhont, D. Verellen, I. Mollaert, V. Vanreusel, and J. Vandemeulebroucke, "RealDRR–Rendering of realistic digitally reconstructed radiographs using locally trained image-to-image translation," *Radiotherapy and Oncology*, vol. 153, pp. 213–219, 2020.

[13] F. Ritter, T. Boskamp, A. Homeyer, H. Laue, M. Schwier, F. Link, and H.-O. Peitgen, "Medical Image Analysis," *IEEE Pulse*, vol. 2, no. 6, pp. 60–70, Nov. 2011, doi: 10.1109/MPUL.2011.942929.

[14] E. Lin and A. Alessio, "What are the basic concepts of temporal, contrast, and spatial resolution in cardiac CT?," *J Cardiovasc Comput Tomogr*, vol. 3, no. 6, pp. 403–408, 2009, doi: 10.1016/j.jcct.2009.07.003.

[15] Y. Guo, J. Chen, J. Wang, Q. Chen, J. Cao, Z. Deng, Y. Xu, and M. Tan, "Closed-loop Matters: Dual Regression Networks for Single Image Super-Resolution," *arXiv:2003.07018 [cs]*, May 2020, Accessed: May 12, 2021. [Online]. Available: http://arxiv.org/abs/2003.07018

[16] L. Xu, X. Zeng, Z. Huang, W. Li, and H. Zhang, "Low-dose chest X-ray image super-resolution using generative adversarial nets with spectral normalization," *Biomedical Signal Processing and Control*, vol. 55, p. 101600, Jan. 2020, doi: 10.1016/j.bspc.2019.101600.
[17] Y. Chen, F. Shi, A. G. Christodoulou, Z. Zhou, Y. Xie, and D. Li, "Efficient and Accurate MRI Super-Resolution using a Generative Adversarial Network and 3D Multi-Level Densely Connected Network," *arXiv:1803.01417 [cs, eess]*, Jun. 2018, Accessed: May 12, 2021. [Online]. Available: http://arxiv.org/abs/1803.01417
[18] C. You, G. Li, Y. Zhang, X. Zhang, H. Shan, S. Ju, Z. Zhao, Z. Zhang, W. Cong, M. W. Vannier, P. K. Saha, and G. Wang, "CT Super-resolution GAN Constrained by the Identical, Residual, and Cycle Learning Ensemble(GAN-CIRCLE)," *IEEE Trans. Med. Imaging*, vol. 39, no. 1, pp. 188–203, Jan. 2020, doi: 10.1109/TMI.2019.2922960.
[19] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 136–144.
[20] F. Yang, H. Yang, J. Fu, H. Lu, and B. Guo, "Learning Texture Transformer Network for Image Super-Resolution," *arXiv:2006.04139 [cs]*, Jun. 2020, Accessed: Feb. 19, 2021. [Online]. Available: http://arxiv.org/abs/2006.04139
[21] J. T. Bushberg and J. M. Boone, *The Essential Physics of Medical Imaging*. Lippincott Williams & Wilkins, 2011.
[22] J. Wang and D. Fleischmann, "Improving Spatial Resolution at CT: Development, Benefits, and Pitfalls," *Radiology*, vol. 289, no. 1, pp. 261–262, Jun. 2018, doi: 10.1148/radiol.2018181156.
[23] M. T. Niknejad, "Contrast resolution | Radiology Reference Article | Radiopaedia.org," *Radiopaedia*. https://radiopaedia.org/articles/contrast-resolution (accessed May 27, 2021).
[24] D. J. Brenner and E. J. Hall, "Computed tomography--an increasing source of radiation exposure," *N Engl J Med*, vol. 357, no. 22, pp. 2277–2284, Nov. 2007, doi: 10.1056/NEJMra072149.
[25] T. D. DenOtter and J. Schubert, "Hounsfield Unit," in *StatPearls*, Treasure Island (FL): StatPearls Publishing, 2021. Accessed: May 26, 2021. [Online]. Available: http://www.ncbi.nlm.nih.gov/books/NBK547721/
[26] M. H. Lev and R. G. Gonzalez, "17 - CT Angiography and CT Perfusion Imaging," in *Brain Mapping: The Methods (Second Edition)*, A. W. Toga and J. C. Mazziotta, Eds. San Diego: Academic Press, 2002, pp. 427–484. doi: 10.1016/B978-012693019-1/50019-8.
[27] M. Touchette, "Artificial X-ray imaging system (AXIS) – design and evaluation on C-arm performance in operating room and educational settings," University of British Columbia, 2017. doi: 10.14288/1.0345636.
[28] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, Art. no. 7553, May 2015, doi: 10.1038/nature14539.
[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.
[30] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556 [cs]*, Apr. 2015, Accessed: May 20, 2021. [Online]. Available: http://arxiv.org/abs/1409.1556
[31] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 1–9. doi: 10.1109/CVPR.2015.7298594.

[32] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.
[33] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 2261–2269. doi: 10.1109/CVPR.2017.243.
[34] C. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2015.
[35] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, Cambridge, MA, USA, Dec. 2014, pp. 2672–2680.
[36] M. Mirza and S. Osindero, “Conditional Generative Adversarial Nets,” *arXiv:1411.1784 [cs, stat]*, Nov. 2014, Accessed: May 21, 2021. [Online]. Available: http://arxiv.org/abs/1411.1784
[37] A. Radford, L. Metz, and S. Chintala, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks,” *arXiv:1511.06434 [cs]*, Jan. 2016, Accessed: May 21, 2021. [Online]. Available: http://arxiv.org/abs/1511.06434
[38] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 2242–2251. doi: 10.1109/ICCV.2017.244.
[39] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein GAN,” *arXiv:1701.07875 [cs, stat]*, Dec. 2017, Accessed: May 21, 2021. [Online]. Available: http://arxiv.org/abs/1701.07875
[40] A. Brock, J. Donahue, and K. Simonyan, “Large Scale GAN Training for High Fidelity Natural Image Synthesis,” *arXiv:1809.11096 [cs, stat]*, Feb. 2019, Accessed: May 21, 2021. [Online]. Available: http://arxiv.org/abs/1809.11096
[41] T. Karras, S. Laine, and T. Aila, “A Style-Based Generator Architecture for Generative Adversarial Networks,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 4396–4405. doi: 10.1109/CVPR.2019.00453.
[42] Z. Wang, J. Chen, and S. C. H. Hoi, “Deep Learning for Image Super-resolution: A Survey,” *IEEE Trans Pattern Anal Mach Intell*, vol. PP, Mar. 2020, doi: 10.1109/TPAMI.2020.2982166.
[43] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, “Loss Functions for Image Restoration With Neural Networks,” *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–57, Mar. 2017, doi: 10.1109/TCI.2016.2644865.
[44] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual Losses for Real-Time Style Transfer and Super-Resolution,” *arXiv:1603.08155 [cs]*, Mar. 2016, Accessed: May 15, 2021. [Online]. Available: http://arxiv.org/abs/1603.08155
[45] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, and Z. Wang, “Photo-realistic single image super-resolution using a generative adversarial network,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.
[46] M. S. M. Sajjadi, B. Schölkopf, and M. Hirsch, “EnhanceNet: Single Image Super-Resolution Through Automated Texture Synthesis,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 4501–4510. doi: 10.1109/ICCV.2017.481.
[47] D. Bahdanau, K. H. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” 2015.

[48] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2*, 2014, pp. 3104–3112.
[49] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention," in *International Conference on Machine Learning*, Jun. 2015, pp. 2048–2057. Accessed: May 28, 2021. [Online]. Available: http://proceedings.mlr.press/v37/xuc15.html
[50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, Dec. 2017, pp. 6000–6010.
[51] M.-T. Luong, H. Pham, and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1412–1421.
[52] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, Jun. 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423.
[53] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-Training," p. 12.
[54] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End Object Detection with Transformers," in *Computer Vision – ECCV 2020*, Cham, 2020, pp. 213–229. doi: 10.1007/978-3-030-58452-8_13.
[55] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv:2010.11929 [cs]*, Oct. 2020, Accessed: May 21, 2021. [Online]. Available: http://arxiv.org/abs/2010.11929
[56] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, "Pre-Trained Image Processing Transformer," *arXiv:2012.00364 [cs]*, Dec. 2020, Accessed: May 21, 2021. [Online]. Available: http://arxiv.org/abs/2012.00364
[57] Z. Zhang, Z. Wang, Z. Lin, and H. Qi, "Image super-resolution by neural texture transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7982–7991.
[58] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," *arXiv preprint arXiv:1704.00028*, 2017.
[59] R. Owens, "Frequency domain methods," *Computer Vision IT412*, Oct. 29, 1997. https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT5/node4.html (accessed May 27, 2021).
[60] R. Bracewell and P. B. Kahn, "The Fourier Transform and Its Applications," *American Journal of Physics*, vol. 34, pp. 712–712, Aug. 1966, doi: 10.1119/1.1973431.
[61] E. O. Brigham, *The fast Fourier transform and its applications | Guide books*. Prentice-Hall, Inc., 1988. Accessed: May 22, 2021. [Online]. Available: https://dl.acm.org/doi/abs/10.5555/47314
[62] W. T. Cochran, J. W. Cooley, D. L. Favin, H. D. Helms, R. A. Kaenel, W. W. Lang, G. C. Maling, D. E. Nelson, C. M. Rader, and P. D. Welch, "What is the fast Fourier transform?,"

Proceedings of the IEEE, vol. 55, no. 10, pp. 1664–1674, Oct. 1967, doi: 10.1109/PROC.1967.5957.
[63] E. Agustsson and R. Timofte, "NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jul. 2017, pp. 1122–1131. doi: 10.1109/CVPRW.2017.150.
[64] A. Clark, "Pillow (PIL Fork) Documentation," p. 265.
[65] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, Nov. 2003, vol. 2, pp. 1398-1402 Vol.2. doi: 10.1109/ACSSC.2003.1292216.
[66] Z. 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*, vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.
[67] F. Jacobs, E. Sundermann, B. De Sutter, M. Christiaens, and I. Lemahieu, "A fast algorithm to calculate the exact radiological path through a pixel or voxel space," *Journal of computing and information technology*, vol. 6, no. 1, pp. 89–94, 1998.
[68] G. W. Sherouse, K. Novins, and E. L. Chaney, "Computation of digitally reconstructed radiographs for use in radiotherapy treatment design," *Int J Radiat Oncol Biol Phys*, vol. 18, no. 3, pp. 651–658, Mar. 1990, doi: 10.1016/0360-3016(90)90074-t.
[69] N. Milickovic, D. Baltast, S. Giannouli, M. Lahanas, and N. Zamboglou, "CT imaging based digitally reconstructed radiographs and their application in brachytherapy," *Phys Med Biol*, vol. 45, no. 10, pp. 2787–2800, Oct. 2000, doi: 10.1088/0031-9155/45/10/305.
[70] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network," *arXiv:1609.05158 [cs, stat]*, Sep. 2016, Accessed: May 15, 2021. [Online]. Available: http://arxiv.org/abs/1609.05158
[71] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual Dense Network for Image Super-Resolution," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 2472–2481. doi: 10.1109/CVPR.2018.00262.
[72] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 286–301.
[73] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, "Second-Order Attention Network for Single Image Super-Resolution," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 11057–11066. doi: 10.1109/CVPR.2019.01132.
[74] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.
[75] W. Zhang, Y. Liu, C. Dong, and Y. Qiao, "RankSRGAN: Generative Adversarial Networks With Ranker for Image Super-Resolution," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), Oct. 2019, pp. 3096–3105. doi: 10.1109/ICCV.2019.00319.
[76] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-Based Super-Resolution," *IEEE Comput. Graph. Appl.*, vol. 22, no. 2, pp. 56–65, Mar. 2002, doi: 10.1109/38.988747.
[77] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, Jun. 2004, vol. 1, p. I–I. doi: 10.1109/CVPR.2004.1315043.

[78] G. Freedman and R. Fattal, “Image and video upscaling from local self-examples,” *ACM Trans. Graph.*, vol. 30, no. 2, p. 12:1-12:11, Apr. 2011, doi: 10.1145/1944846.1944852.
[79] L. Sun and J. Hays, “Super-resolution from internet-scale scene matching,” in *2012 IEEE International Conference on Computational Photography (ICCP)*, Apr. 2012, pp. 1–12. doi: 10.1109/ICCPhot.2012.6215221.
[80] H. Zheng, M. Guo, H. Wang, Y. Liu, and L. Fang, “Combining Exemplar-Based Approach and learning-Based Approach for Light Field Super-Resolution Using a Hybrid Imaging System,” in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, Oct. 2017, pp. 2481–2486. doi: 10.1109/ICCVW.2017.292.
[81] J. Kim, J. K. Lee, and K. M. Lee, “Accurate Image Super-Resolution Using Very Deep Convolutional Networks,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 1646–1654. doi: 10.1109/CVPR.2016.182.
[82] H. Zheng, M. Ji, H. Wang, Y. Liu, and L. Fang, “Crossnet: An end-to-end reference-based super resolution network using cross-scale warping,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 88–104.
[83] H. Gothwal, S. Kedawat, and R. Kumar, “Cardiac arrhythmias detection in an ECG beat signal using fast fourier transform and artificial neural network,” *Journal of Biomedical Science and Engineering*, vol. 4, no. 4, Art. no. 4, Apr. 2011, doi: 10.4236/jbise.2011.44039.
[84] M. Mironovova and J. Bíla, “Fast fourier transform for feature extraction and neural network for classification of electrocardiogram signals,” in *2015 Fourth International Conference on Future Generation Communication Technology (FGCT)*, Jul. 2015, pp. 1–6. doi: 10.1109/FGCT.2015.7300244.
[85] Z. Zhang, Y. Wang, and K. Wang, “Fault diagnosis and prognosis using wavelet packet decomposition, Fourier transform and artificial neural network,” *J Intell Manuf*, vol. 24, no. 6, pp. 1213–1227, Dec. 2013, doi: 10.1007/s10845-012-0657-2.
[86] M. Mathieu, M. Henaff, and Y. LeCun, “Fast Training of Convolutional Networks through FFTs,” *arXiv:1312.5851 [cs]*, Mar. 2014, Accessed: May 19, 2021. [Online]. Available: http://arxiv.org/abs/1312.5851
[87] H. Pratt, B. Williams, F. Coenen, and Y. Zheng, “FCNN: Fourier Convolutional Neural Networks,” in *Machine Learning and Knowledge Discovery in Databases*, Cham, 2017, pp. 786–798. doi: 10.1007/978-3-319-71249-9_47.
[88] S. Lin, N. Liu, M. Nazemi, H. Li, C. Ding, Y. Wang, and M. Pedram, “FFT-based deep learning deployment in embedded systems,” in *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, Mar. 2018, pp. 1045–1050. doi: 10.23919/DATE.2018.8342166.
[89] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville, “On the spectral bias of neural networks,” in *International Conference on Machine Learning*, 2019, pp. 5301–5310.
[90] K. Xu, M. Qin, F. Sun, Y. Wang, Y.-K. Chen, and F. Ren, “Learning in the frequency domain,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1740–1749.
[91] J. Li, S. You, and A. Robles-Kelly, “A frequency domain neural network for fast image super-resolution,” in *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–8.
[92] K. Zhang, W. Zuo, and L. Zhang, “Learning a Single Convolutional Super-Resolution Network for Multiple Degradations,” *arXiv:1712.06116 [cs]*, May 2018, Accessed: May 15, 2021. [Online]. Available: http://arxiv.org/abs/1712.06116

[93] S. A. Hussein, T. Tirer, and R. Giryes, “Correction Filter for Single Image Super-Resolution: Robustifying Off-the-Shelf Deep Super-Resolvers,” *arXiv:1912.00157 [cs, eess]*, May 2020, Accessed: May 15, 2021. [Online]. Available: http://arxiv.org/abs/1912.00157

[94] S. K. Lam, A. Pitrou, and S. Seibert, “Numba: a LLVM-based Python JIT compiler,” in *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, New York, NY, USA, Nov. 2015, pp. 1–6. doi: 10.1145/2833157.2833162.

[95] Y. Wang, Z. Lin, X. Shen, R. Mech, G. Miller, and G. W. Cottrell, “Event-Specific Image Importance,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 4810–4819. doi: 10.1109/CVPR.2016.520.

[96] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Sep. 1999, vol. 2, pp. 1150–1157 vol.2. doi: 10.1109/ICCV.1999.790410.

[97] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, “ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 3462–3471. doi: 10.1109/CVPR.2017.369.

[98] W. Birkfellner, R. Seemann, M. Figl, J. Hummel, C. Ede, P. Homolka, X. Yang, P. Niederer, and H. Bergmann, “Fast DRR Generation for 2D/3D Registration,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2005*, vol. 3750, J. S. Duncan and G. Gerig, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 960–967. doi: 10.1007/11566489_118.