5-2023

Nonlinear Mathematical Transformations for Improved Image and Signal Recovery Using Artificial Neural Network

Haoran Chang
Florida Institute of Technology

Nonlinear Mathematical Transformations for Improved Image and Signal Recovery Using Artificial Neural Network

by

Haoran Chang

Master of Science
Computer Science
Florida Institute of Technology
2017

Bachelor of Computer Science
Department of Computer Science
Shandong University
2012

A dissertation
submitted to the College of Engineering and Science
at Florida Institute of Technology
in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy
in
Computer Science

Melbourne, Florida
May, 2023

We the undersigned committee
hereby approve the attached dissertation

Nonlinear Mathematical Transformations for Improved Image and Signal Recovery
Using Artificial Neural Network by Haoran Chang

Debasis Mitra, Ph.D.
Professor
Computer Engineering and Sciences
Major Advisor

Samuel P. Kozaitis, Ph.D.
Professor
Computer Engineering and Sciences

Eraldo Ribeiro, Ph.D.
Associate Professor
Computer Engineering and Sciences

Marius C. Silaghi, Ph.D.
Professor
Computer Engineering and Sciences

Philip J. Bernhard, Ph.D.
Associate Professor and Department Head
Computer Engineering and Sciences

Abstract

Title:

Nonlinear Mathematical Transformations for Improved Image and Signal Recovery Using Artificial Neural Network

Author:

Haoran Chang

Major Advisor:

Debasis Mitra, Ph.D.

Medical imaging plays a vital role in modern healthcare, enabling clinicians to diagnose and treat a range of conditions. However, image acquisition and processing can be challenging because they can often be hindered by motion blurring, leading to inaccurate results. To address that, this dissertation proposes a novel approach based on nonlinear mathematical transformations and artificial neural networks (ANN). The dissertation begins with an introduction to Nuclear Medicine and the problem of motion blur in image reconstruction. A background on Medical Imaging techniques, including the Radon transform and Image Reconstruction methods such as Filtered Back Projection and Iterative Reconstruction methods are presented. The ANN is introduced, including the Fully Connected Neural Network (FCN) and Convolutional Neural Network (CNN). Then related work is presented, including previous studies on Deep Learning for Medical Imaging and motion correction techniques, such as image denoising

and motion correction. Preliminary experiments are conducted to test the viability of using Deep Learning techniques for parameter prediction, synthetic object reconstruction, and motion blur handling. The main contribution of this dissertation is the proposed ANN model for reconstruction from motion blurred sinogram data, and moreover using zero-shot learning to reconstruct the motion-free image. The methodology is described in detail, including the use of a CNN with the Self-Attention mechanism. Experimental results demonstrate the effectiveness of the proposed method in producing accurate image reconstructions, with improved image quality and reduced motion blur. Overall, this dissertation presents a novel approach to image reconstruction for Nuclear Medicine Imaging, using Deep Learning techniques to address the problem of motion blur in sinogram data. The proposed method has the potential to improve diagnostic accuracy and enhance patient care in clinical settings.

Table of Contents

List of Figures

List of Tables

Acknowledgements

Research reported in this publication was supported by the National Institute of Biomedical Imaging And Bioengineering of the National Institutes of Health (NIH) under Award Number R15EB030807. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

I would also like to express my sincere appreciation to my advisor Dr. Debasis Mitra, whose mentorship and guidance have been invaluable throughout my academic journey. His expertise, patience, and commitment to my success have been instrumental in shaping my research and personal growth.

I would like to express my sincere gratitude to the committee members, Dr. Eraldo Ribeiro, Dr. Marius C. Silaghi, and Dr. Samuel P. Kozaitis, for their valuable guidance and support throughout the research process.

I am deeply thankful to Dr. Rostyslav Boutchko for his invaluable assistance and expertise, which greatly contributed to the success of my research.

I would also like to extend my appreciation to the esteemed faculty members at UCSF, namely Drs. Uttam Shrestha, Grant T. Gullberg, and Youngho Seo, for their valuable insights and support.

Furthermore, I would like to acknowledge the contributions of the faculty members at Cardiff University, Drs. Rhodri L. Smith and Stephen Paisey, for their valuable input and guidance.

I am grateful to all the individuals mentioned above for their significant contributions and support, which have been instrumental in the successful completion of this research project. A support from the Supercomputing Wales project, partly funded by the European Regional Development Fund (ERDF) via the Welsh Government, is also acknowledged.

We thank the Health First corporation's Holmes Regional Medical Center in Melbourne, Florida for supporting our work by providing the human data.

We also acknowledge kind support of a TITAN Xp Graphics Card from NVIDIA.

We are grateful to Dr. Paul Segars from Duke University Medical Center for providing help in using MOBY.

Dedication

I would like to dedicate this dissertation to my parents, whose unwavering support and encouragement have been the driving force behind my academic pursuits. Their love and sacrifices have made it possible for me to pursue my dreams, and I am forever grateful for their guidance and care.

Thank all of you for being a constant source of inspiration and guidance. This accomplishment would not have been possible without your unwavering support.

Chapter 1

Introduction

Many years ago, when doctors wanted to understand the internal condition of a patient, in addition to directly opening up the patient's body, they could only rely on palpation, but both of these methods have a certain risk. The former one would seriously injure the patient, and the later one might not provide enough information for the diagnosis. In 1895, when German physicist Wilhelm Röntgen discovered X-rays, it opened a new chapter in medical imaging. This provided a non-invasive way to help doctors probe the inside of the body. After that, researchers developed many other medical imaging modalities. In this chapter, we will first briefly introduce the nuclear medicine imaging, then state the problems in image reconstruction, and list the main contributions in our studies.

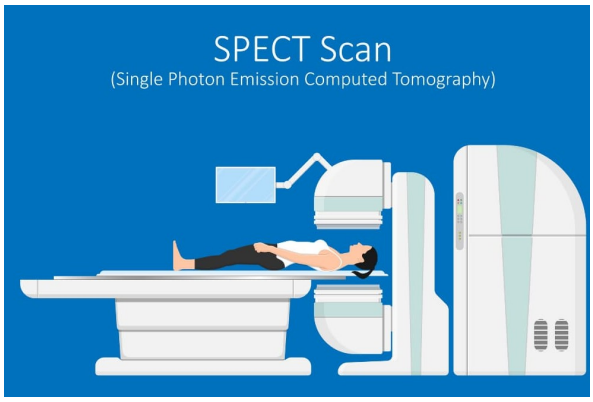
1.1 Nuclear Medicine Introduction

Nuclear medicine is a branch of medicine that focuses on medical imaging. It is a field that uses nuclear techniques for disease diagnosis, treatment, and research, a product of the combination of medicine and modern science such as nuclear technology,

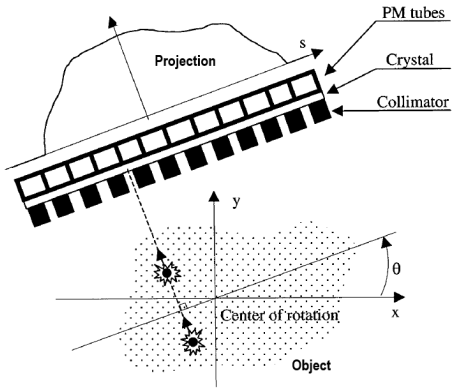
electronic technology, computer science and so on. Nuclear medicine is an application with radioisotopes, nuclear radiation from radioisotopes, and ray beams generated by some special accelerators. In medicine, radioisotopes and nuclear radiation can be used for diagnosis, treatment and medical research. In pharmaceuticals, they can be used to study the principles of drug effects, measuring the activity of drugs, and drug analysis. There are many kinds of techniques of nuclear imaging: X-ray computed tomography (XCT), positron emission tomography (PET), nuclear magnetic resonance (NMR), single photon emission computed tomography (SPECT).

From the 1970s, nuclear imaging has made a breakthrough, due to the development of singe photon emission computed tomography (SPECT) and positron emission tomography (PET), as well as the innovation of radio pharmaceutical. Nuclear imaging, CT, MRI, and ultrasound are complementary imaging modalities that can greatly enhance disease diagnosis and research. Through this, the level of the disease research and diagnosis are greatly improved. Therefore, nuclear imaging is a very popular and important part of clinical diagnosis based on imaging..

Nuclear medicine imaging is a medical imaging technique based on the radionuclide tracers. Nuclear imaging techniques using CT is also called emission computed tomography (ECT), because they are imaging by collecting the gamma ray transmitted from the patient. According to the type of radionuclide used in ECT, there are two categories, which are also our main research aspects: single photon emission computed tomography (SPECT) and positron emission tomography (PET). For both of SPECT and PET, patients need to take the injection of different tracers. The type of tracer used depends on which organ or tissue that the doctor wants to examine. Because the radioisotopes are unstable, they undergo radioactive decay and emit gamma rays. Though both SPECT and PET use a gamma camera to collect rays, there are many differences in details between them.



(a) A SPECT scan machine [8]. Picture from a blog in AskApollo, "What is a SPECT Scan Commonly Used For?"



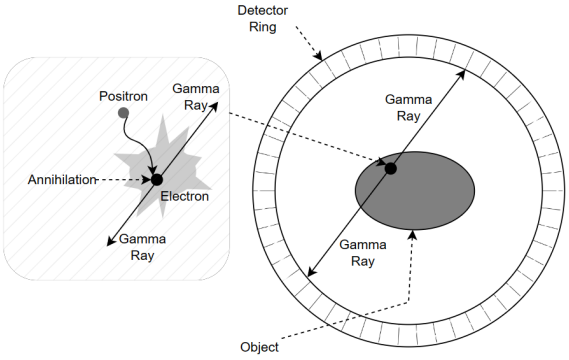
(b) The principle of SPECT data acquisition and geometry [9].

Figure 1.1

In SPECT, the tracer emits the gamma rays directly during decay. Note that decay or radioactive decay is a phenomenon in which unstable atomic nuclei emit ionizing radiation and switch their nuclear composition or energy levels after a certain period of time. Gamma decay is one type of radioactive decay that emits gamma rays. The gamma camera probe detects gamma photons from one projection line (ray). The measurement values in each point represent the sum of radioactivity in this line. These sensitive points in a same line on the camera can detect the activity of radiopharmaceutical on a slice of human body. The output is called the one-dimensional projection (Projection). Due to the collimators, only the projection lines that are parallel to each other and perpendicular to the detector can be collected, so it is called a parallel beam. When the detector rotates, people can obtain the projections of different angles. Because the distance between detector and emission photon is unknown at one angle, it needs to be viewed from different angles to know the body structure in perpendicular direction. It has been proven that if all the projections of all the angles are known, the tomographic image can be computed. Fig. 1.1 shows an example of SPECT Scan and its physics.

PET uses a tracer that first emits a positron, which travels a short distance (in

(a) A PET scan machine. Picture from the National Health Service (NHS) website [10].



(b) Ring of PET scanner and the event.

Figure 1.2

millimeters) before colliding with an oppositely charged electron. This collision results in the emission of two gamma photons in opposite directions, a process known as annihilation. When two detectors detect these gamma photons, the path between them is referred to as a line of response (LOR), and the detection is called a "coincidence" event. Because these events provide more precise localization information, PET can offer better contrast and spatial resolution than SPECT. However, PET is much more expensive than SPECT due to the difficulty in obtaining the PET tracer. In contrast to SPECT, PET machines do not require collimators; instead, they use a ring of detectors to collect rays from different angles.

The raw data collected by SPECT or PET machines is known as a *sinogram*. An example of a 2D image and its corresponding sinogram is shown in Figure 1.3. The 2D object in the figure is called the Shepp-Logan phantom [1], which is used as a standard test image for modeling a human head. The main objective of nuclear imaging is to reconstruct the object as accurately as possible from its sinogram, which is an inverse problem. The reconstruction process involves using mathematical methods to calculate the causal factors (original object) from a series of observed effects (sinogram). More details about the image reconstruction methods will be discussed in Chapter 2.

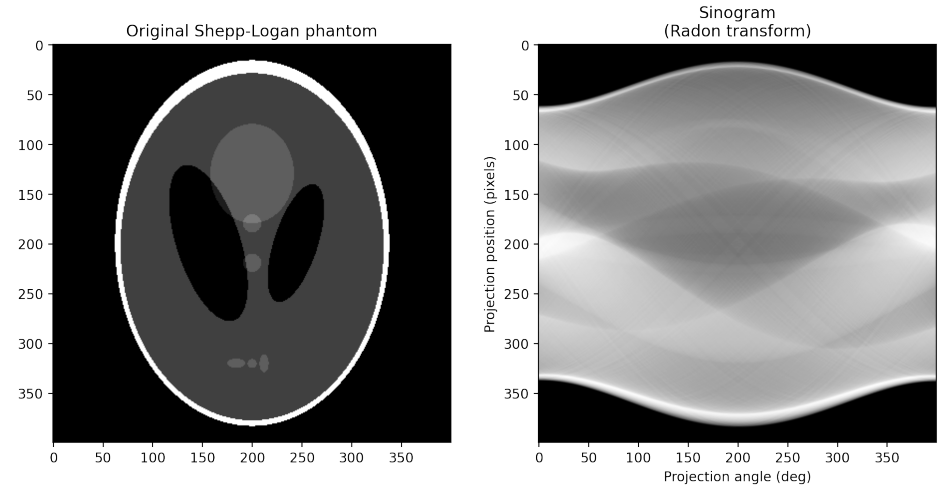


Figure 1.3: An example of the original 2D object and its sinogram. ***Left:*** Shepp-Logan phantom, which is a standard test image and serves as the model of a human head [1]. ***Right:*** the corresponding sinogram of the Shepp-Logan phantom using 0-180 degrees. Note that in the sinogram image, one of the axes is the angle of projection, while both of the axes are the pixel position in the original image.

1.2 Problem Overview

Due to its rapid development, deep learning has the potential to provide solutions to problems in many areas, including medical imaging. However, image reconstruction is a complex problem that poses several challenges when applying deep learning methods. Here, we discuss some of the issues and challenges involved in using deep learning for image reconstruction:

- **Difference space**: There is a difference in space between the sinogram, which is in the projection space, and the object, which is in the ordinary 2D or 3D space. Although artificial neural networks (ANNs) can estimate or approximate complex functions, it is essential to investigate their ability to estimate a mathematical transformation from one space to another. As the reconstruction problem is ill-posed, the precision of the ANN output should be carefully evaluated.

- **Motion**: Subject movement during the scan, such as breathing and heartbeats, can introduce strong artifacts and pose challenges for medical diagnosis. Tradi-

tionally, this problem is solved by bringing extra steps to suppress motion blur. However, this can cause other small problems. It is possible to use ANNs to reconstruct the motion-free image using the raw motion-blurred data directly, without processing.

- **Data amount limitation**: To train a deep learning model, a large amount of data is required. However, the availability of medical data is limited due to privacy concerns or the rarity of certain diseases. This data amount limitation can pose a challenge for training deep learning models for image reconstruction in medical imaging.

1.3 Main Objectives

Our main objectives in this dissertation are:

1. We explore the capability of ANN for different transformations, such as using ANN to reconstruct the synthetic objects, predict the inverse discrete Fourier transform, estimate certain parameters given the attenuated objects or the motion blurred objects, synthetic object reconstruction, and motion correction.

2. We developed a convolutional encoder-decoder with self-attention components (CEDA) to reconstruct the motion-free image given the motion-corrupted sinogram. We test our model on the simulation data (using affine motion) and the real human data, and compared against the existing ANN models in the literature.

3. Considering the situation of the limited data, we use only the simulated mouse data to train our model, and then validate with similar real mouse data. And the results show our model performs better than the traditional reconstruction method.

1.4 Chapter Contents

There are seven chapters in this dissertation. In the first chapter, we briefly explain the nuclear imaging and background to our research. The second chapter demonstrates some basic methods that are used in image reconstruction and deep learning. The third chapter reviews the related work. The forth chapter presents some of our preliminary works with some deep learning methods. The fifth chapter displays the motion-free reconstruction directly from motion blurred sinogram using neural network. And the six chapter gives another project that reconstructing the real mouse data by a deep learning model which is trained with phantom data. In the last chapter, I conclude the the studies, discuss the limitation and elaborate the future work.

Chapter 2

Background

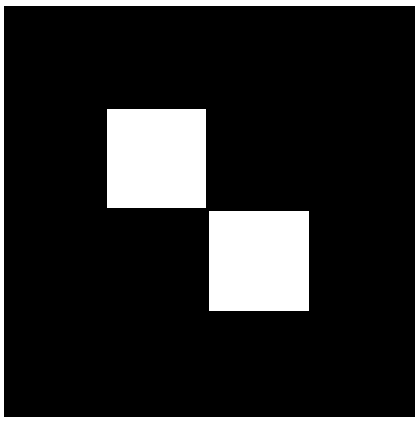
2.1 Medical Imaging

Generally speaking, medical imaging is the technology and process obtaining the 2D or 3D images of interior of the human or animal body by a non-invasive way. These images can show the tissue and organs of the body, which are used for medical treatment or medical research. Technically, medical imaging uses a certain kind of wave or ray to penetrate the patient. The projections will be generated after these waves or rays go through the patient, and collected by some specific devices. Because of the difference of the tissues, this projection can show the structure inside the patient. In order to reconstruct the 3D image of the patient, a number of projections from different angles will be collected. This method is called tomography, which has been widely used in many areas, as well as medical imaging. Tomography reconstruction (TR) is the technique to reconstruct the image from these projections. TR is an inverse problem, thus the final resulting image (characteristics of living tissue) is inferred through the observed image signal (which is obtained by the machine). Therefore, most of the techniques are trying to obtain a better reconstruction performance. In

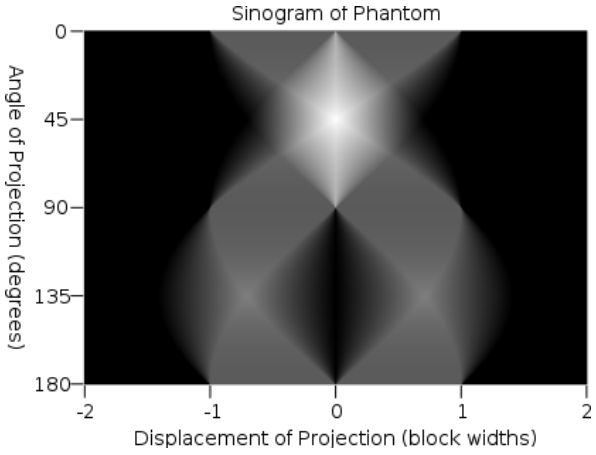
the next subsections, we will explain how the raw data is collected, and what the reconstruction methods are.

2.1.1 Radon Transform

As we mentioned before, TR is an inverse problem. The main challenge of it is to generate the an estimated volume of a targeted object from some discrete projections. The mathematical foundations of TR is Radon transform [11]. Radon transform as well as its inverse transform was introduced by Johann Radon in 1917. By convention, the result of the Radon Transform is called "sinogram", since the Radon transform of a point which is not the center is a sinusoid. The Radon transform of some small objects looks like several sine waves with different amplitudes and phases overlapping.



(a) Original function. White region is one, black region is zero.



(b) Radon transform of fig 2.1a.

Figure 2.1: The Radon transform(right) of an object(left). The result is called sinogram. Images from Wikipedia: Radon Transform.

Radon transform is a kind of integral transform. If the function f represents an unknown density, doing a Radon transform on f is equivalent to obtaining the signal of the projection of f. It's applied in computed tomography [12] [13], electron microscopy of macromolecular assemblies [14], and so on.

If we say a density function $f(\mathbf{x}) = f(x, y)$ is a compact support with the domain $\mathfrak{R}^2$, which means it is a closed and bounded subset of $\mathfrak{R}^2$. Let $\mathcal{R}$ is the operator of Radon transform, then $\mathcal{R}f$ is a line L defined in $\mathfrak{R}^2$:

$$\mathcal{R}f(L) = \int_L f(x, y)ds$$

Where ds is the differential element of the line L. Radon transform can also be defined in n-dimensional Euclidean space $\mathfrak{R}^n$. That is, $\mathcal{R}f$ is on the space Σ_n of the hyperplanes in $\mathfrak{R}^n$:

$$\mathcal{R}f(\xi) = \int_\xi f(\mathbf{x})|d\sigma(\mathbf{x})|$$

for $\xi \in \Sigma_n$.

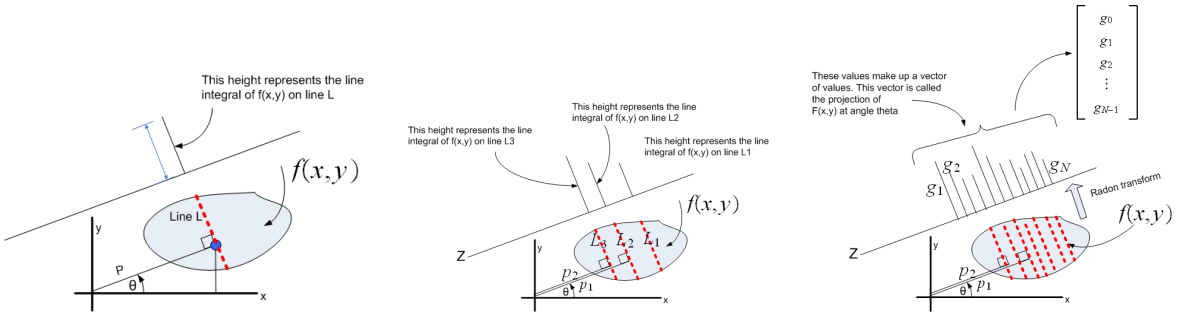
Fig shows how Radon transform obtains the projection of a certain angle. On the other hand, we know the line equation can be: $y = ax + b$, where a is the slope and b is the intercept. It can also be written as: $L(p, \theta) := \{(x, y) \mid x\cos\theta + y\sin\theta = p\}$, where p is the distance from the origin to the line L, θ is the angle the normal vector to L makes with the x axis. Therefore, Radon transform can be written as:

$$g(p, \theta) = \mathcal{R}f(p, \theta) = \int_{y=-\infty}^{+\infty} \int_{x=-\infty}^{+\infty} f(x, y)\delta(x\cos\theta + y\sin\theta - p)dxdy$$

$\delta(\mathcal{X})$ is the delta function.

A common solution for the inverse of Radon transform is the back projection algorithm (sometimes called duel Radon transfrom [15]).

$$f(x, y) = \int_{\theta=0}^{\pi} g(x\cos\theta + y\sin\theta)d\theta$$



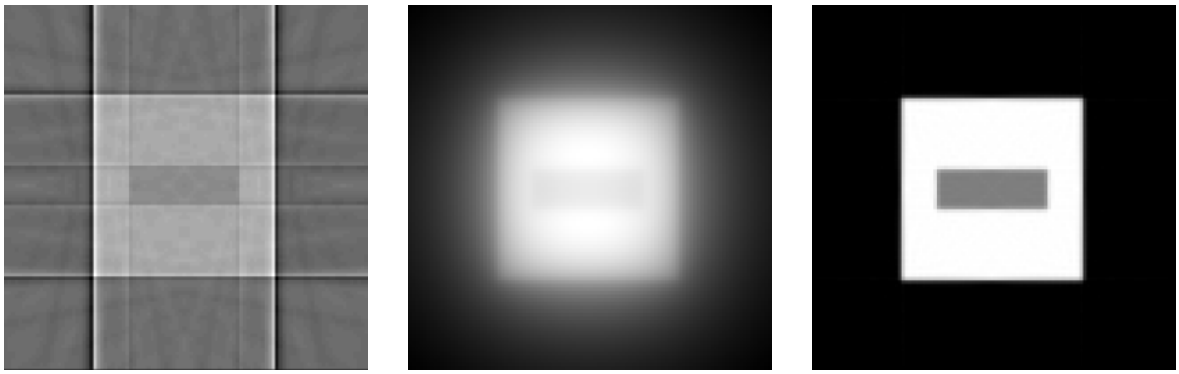
(a) One line. (b) Two lines. (c) Multiple lines.

Figure 2.2: How Radon transform works for a certain line(s). p is the distance from the origin to a certain line. θ is the angle the normal vector to L makes with the x axis. [2].

In reality, we use a discrete approximation:

$$f(x,y) \approx \Delta\theta \sum_{i=0}^{N-1} g(x\cos\theta_i + y\sin\theta_i)$$

Obviously, the more projections are used, the more accurate the reconstruction image is(fig 2.3a and fig 2.3b), but the reconstruction image will be heavily blurred (fig 2.3b). One solution is using filters. Filtering can be used before the back projection or after, but today most of the algorithms do that before.

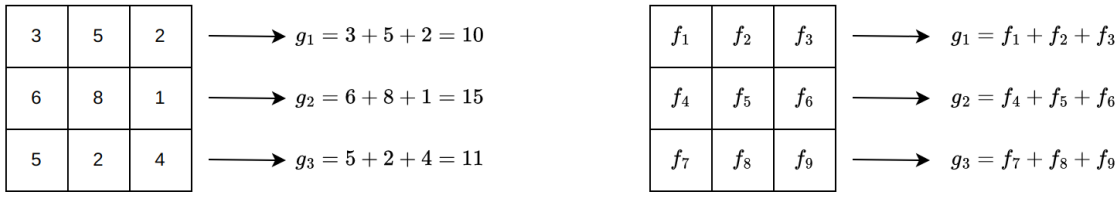


(a) Reconstruction using 18 projections. (b) Reconstruction using all the projections. (c) Filtered back projection. Using ramp filter

Figure 2.3: Reconstruction using back projection [3].

2.1.2 Image Reconstruction

Radon transform and its inverse reveal the mathematical meaning of the projection. But in real life, the data is finite and discrete. The (forward) projection process is described in fig 2.4a. More general, if we use variables instead of actual values, we can get fig 2.4b. g_i is the projection value, which is a certain pixel in the sinogram. f_j is the voxel value of the object.



(a) An example of forward projection.

(b) A more general form of the projection.

Figure 2.4

Let's see the three equations in fig 2.4b. Each g_i is only related with few voxels (f_j). We can add other voxels with zero coefficients so that the equations are still hold (eq 2.1 2.3).

$$g_1 = f_1 + f_2 + f_3 = f_1 + f_2 + f_3 + 0 \cdot f_4 + 0 \cdot f_5 + 0 \cdot f_6 + 0 \cdot f_7 + 0 \cdot f_8 + 0 \cdot f_9 \quad (2.1)$$

$$g_2 = f_4 + f_5 + f_6 = 0 \cdot f_1 + 0 \cdot f_2 + 0 \cdot f_3 + f_4 + f_5 + f_6 + 0 \cdot f_7 + 0 \cdot f_8 + 0 \cdot f_9 \quad (2.2)$$

$$g_3 = f_7 + f_8 + f_9 = 0 \cdot f_1 + 0 \cdot f_2 + 0 \cdot f_3 + 0 \cdot f_4 + 0 \cdot f_5 + 0 \cdot f_6 + f_7 + f_8 + f_9 \quad (2.3)$$

Say A is the coefficient matrix, g is the vector of stacking all g_i, and f is the vector of all f_j. Then the vector g can be evaluated by the product of coefficient matrix A and

the object vector f:

$$\begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \end{bmatrix}$$

The more projections of different angles that the sinogram has, the more elements in the vector g_i. In general, we have:

$$g = Af \tag{2.4}$$

In this example, we only use a single angle (90 degree) to show the projection. Therefore, there are only 3 g_i shown here. The more number of projections is used, the more g_i will be in vector g, and the more number of rows matrix A has. Still, we can get $g = Af$. As we can see, coefficient matrix A is not related to the value of g_i and f_j. It is defined by the model, or the projection system. When the system is fixed, we can compuate A without knowing any actual g_i or f_j values. There is one problem when we use our A for computation. Currently, all the elements in A are binary (only 0 or 1), which means at this angle, a certain voxel f_j will project to a certain sinogram pixel g_i or not. But it is impossible in real life. Because of the reasons like random noise, scattering, system errors and so on, f_j may project to other sinogram pixels partially.

So the elements in A are the fraction numbers between 0 1.

In $g = Af$, g is the sinogram which is known, A is the system matrix which can be calculated. So the reconstruction problem is, given g and A, how to find f. Ideally, we can compuate the inverse of A, that is A^{-1}, and then get f by $f = A^{-1}g$. But this method has some fatal problem in practice: (1) Calculating A^{-1} is unfeasible or at least high intensive since A^{-1} is huge; (2) A^{-1} may not exist because A^{-1} may not be square or it is singular; (3) A^{-1} can be ill-conditioned, so that a change in g will cause great differences in f. In the following sections, several common reconstruction method will be demonstrated including FBP, CG, and MLEM.

2.1.2.1 Filtered Back Projection

Fig 2.3a and 2.3b show an example of the backprojection. Ideally, backprojection can be defined by:

$$bp(x, y) = \int_0^{\pi} g(s, \theta) d\theta$$

Same as Radon transform, the projections only covers from 0 to π radians, because another half of the rotation (π to 2π) will get the same values as the first half in the ideal situation. Moreover, in discrete domain, which is closer to practice, the formula of the backprojection can be written as:

$$bp_D(x, y) = \sum_{k=1}^{n} g(s_k, \theta_k) \Delta\theta$$

where n is the number of discrete projections, and $\Delta\theta$ is the angle interval between two successive projections ($\Delta\theta = \pi/n$), s_k is the $k - th$ location along the detector and θ_k is the $k_t h$ projection. For example, in fig 2.1b, s_k shows the x-coordinate in the sinogram, while θ_k is the y-coordinate.

Let's see an example about how backprojection works. Suppose we have an 3×3

object, the pixel value within the object is unknown. Fig 2.5 shows an example of the discrete backprojection computing given two projections which are at angle 0 and $\pi/2$.

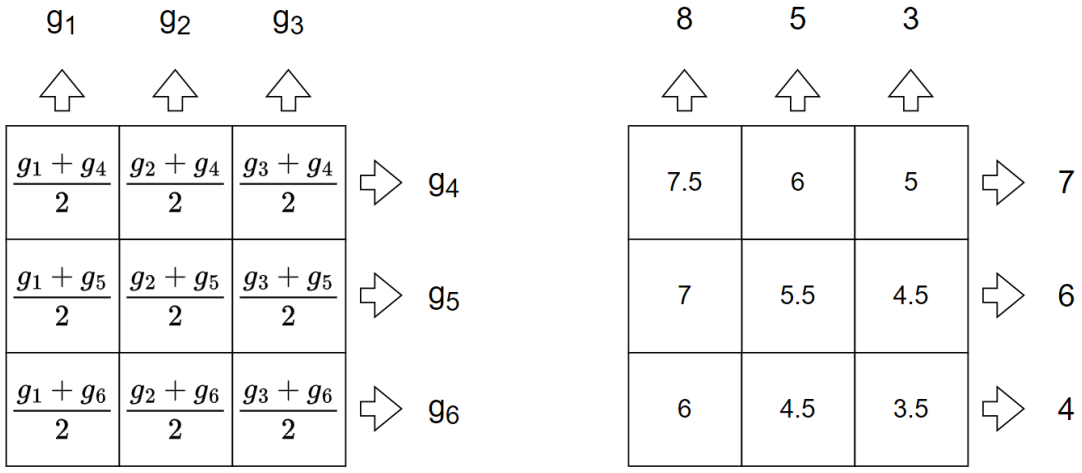


Figure 2.5: The process of how to calculate the backprojection given the projections. The left one shows the algebra and the right one is an example of some real numbers.

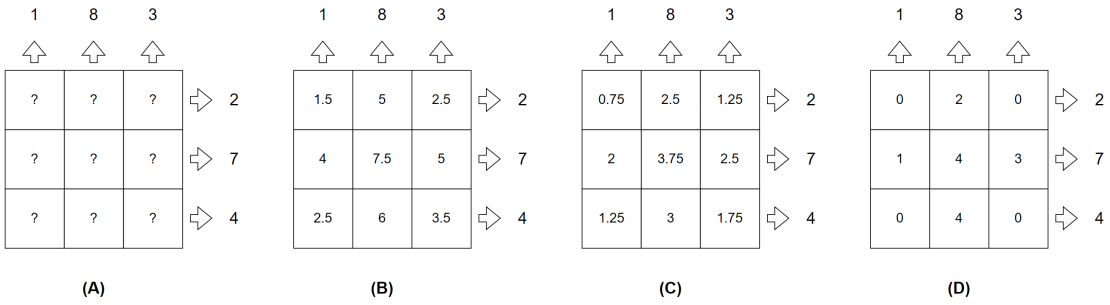


Figure 2.6: A simple example to show that how the blur is introduced by the backprojection process. (A) Given the projections but pixel value of the image is unknown. (B) Backprojection to find the all 9 pixel values. (C) Final result of the backprojection. (D) Actual pixel value of the image. See the difference between any two pixels in C and D. One can notice that the difference in C is lower than in D.

Backprojection may introduce blur. Fig 2.6 shows an example of how a blurred backprojected image (C in fig 2.6) is generated. Note that the absolute difference of any pixel values in C is lower than in D. Thus, the pixel values in C are more closer

to each other, and it makes image be blur. On the other hand, we can see the star artifacts when using limited number of projections backprojection (fig 2.7).

Technically, star artifacts can be suppressed by using more number of projections for the backprojection. If we can use infinite projections, we can perfectly reconstruct the original image. But it is impossible. Normally, people prefer to use a convolution filter to remove blurring. For example, we can give a small weight for those low frequency components so that the the high frequency components, which are most likely to be the edges, are emphasized. There are two ways to do this filtering. One can backproject the sinogram and then filter the backprojection. While another way, which is more common in real life, is to filter the projections first then backproject the filtered projections. People called the later method filtered backprojection (FBP [16]).

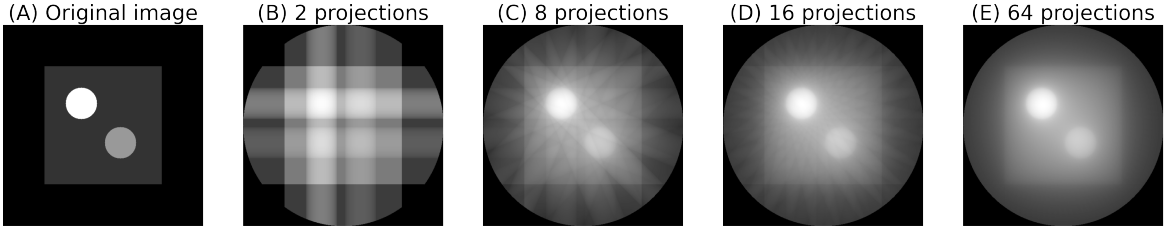


Figure 2.7: Star artifacts in backprojection. (A) Original image. (B-E) backprojection using 2, 8, 16, 64 number of projections which are equally distributed between 0 to π (180 degree). We can clearly see the star artifacts at the beginning. During the number of projections increasing, star artifacts decreases.

As we know, frequency is defined by the repeating times of an event happens during a certain period. High frequency means a large amount of occurrences of this event during a unit period. Therefore, the frequency of an image means the change of the pixel value in any direction within a certain distance (that's why image frequency is also called spatial frequency). High frequency in a image means the pixel value change rapidly, which is usually the edge of an object. To filter an image, we normally use Fourier transformation (FT) to spilt it into different components, then apply the filter to weight these components, and finally sum up these weighted components to get the

image back. Fig 2.8A shows some of the common filters used in FBP. Fig 2.8 B-E shows the results of FBP using different filters. The original image is the same as fig 2.7(A). The sinogram of this image to do FBP contains 64 projections which are evenly distributed between 0 to π.

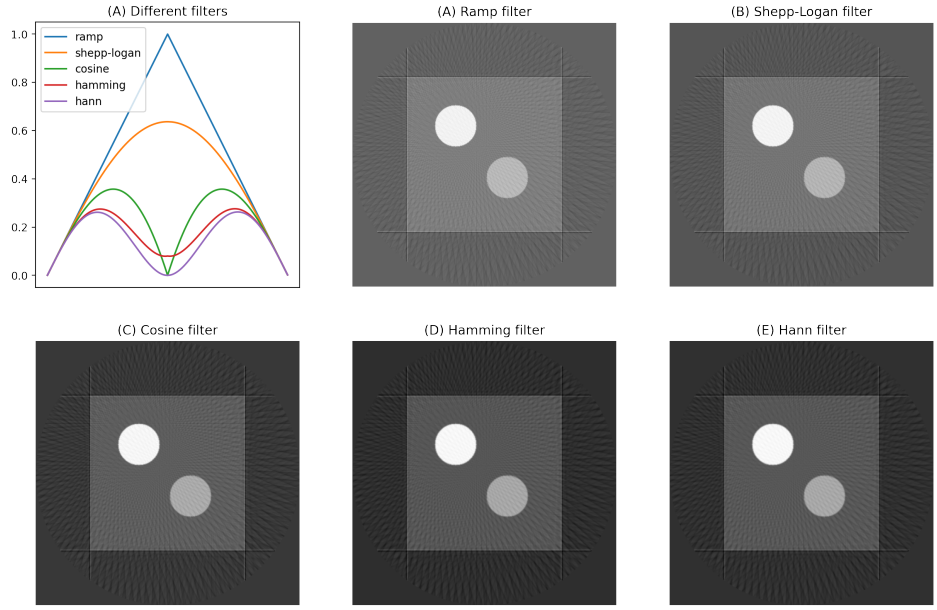


Figure 2.8: FBP using different kind of filters. (A) Original image. (B-E) FBP using Ramp, Shepp-Logan, Cosine, Hamming, and Hann filter. The sinogram used here has 64 projections which are evenly distributed between 0 to π.

We can apply a filter that reduces the low frequency components of the sinogram (like Ramp filter). Thus, the high frequency components are enhanced. That means the edges of the reconstructed image can be sharpened. However, it is unfortunate that much image noise is in high frequency. An alternative choice is to use a filter that can restrain both lowest and highest frequency components. Now there is a trade-off: we can use a stronger filter to reduce more noise, but we will also lose more information of the image.

2.1.2.2 Iterative Reconstruction Method

Algebraic Reconstruction Technique

FBP derives the results directly from the projections. So the process of FBP can be very fast. There is another type of reconstruction method which is slower than FBP, but has a better quality. It is called iterative reconstruction method. Recall that our goal is to solve $g = Af$. Iterative method first initializes an estimated f, measures the difference between the estimated f and the given sinogram g, and then tries to improve the estimation based on the difference in the next iteration. A simple approach is called the algebraic reconstruction technique (ART eq. (2.5)).

$$f_j^{(k+1)} = f_j^{(k)} + \frac{g_i - \sum_{j=1}^{N} f_{ij}^{(k)}}{N} \tag{2.5}$$

In equation (2.5), $f_j^{(k)}$ is the estimation at the k-th iteration; $f_j^{(k+1)}$ is the estimation of next iteration ($(k+1)$-th); g_i is the projection along the ray i; $\sum_{j=1}^{N} f_{ij}^{(k)}$ is the sum of all the pixel values along ray i of the k-th estimation; N is the number of pixels along ray i. Fig 2.9 shows an example of ART application.

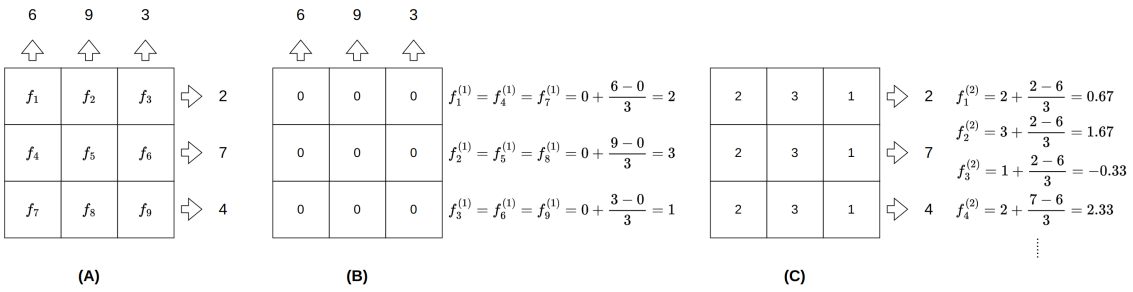


Figure 2.9: An example of how ART reconstructs the image. (A) Unknown image with two projections at angle 0 and angle $\pi/2$. (B) Initialize all the pixel values to be zero, then calculate the the first estimation using the projection at angle 0. (C) Calculate the second estimation using the projection at angle $\pi/2$. Here we only show the calculation of 4 pixels. Rest of the pixels are computed with similar way.

Gradient

There is another approach to find the best estimation of f. Suppose we initialize a large number of different images. If we compute the projections of them, then compare these projections with the measured projection by subtraction, we can get the differences between our estimations and the measured sinogram. Some of them may have large differences, while others may have small differences. Our target is to find the one that has the smallest difference with the measured projection.

However, to create thousands or millions of different images and compute the differences is monotonous and prolonged. We prefer a clever way to find the image that has the minimal difference. Back to the numerous differences we mentioned in the previous paragraph. If we plot them (suppose they can be plotted in the 3D coordinate system), we shall visualize a terrain with a number of valleys and peaks, while valleys mean small differences and peaks mean high differences. Starting from a random point on this terrain, if we want to go to the lowest point (valley), we may look around, find the direction to a lower point, and go there. Repeating this process, we can finally reach the lowest point. We always want to reach the lowest point as fast as possible, hence, the direction and the step length we walk must be carefully chosen. Intuitively, go down the steepest slope can help us reach the lowest point faster. Therefore the direction should always point to the steepest slope. Besides, the step length is specially designed, so that we can stop before ascending. This algorithm is called gradient algorithm.

In general, the reconstruction problem can be described as: find the best solution (estimation) f so that the difference between the estimated projection and given projection is minimized. Given the estimated image f, we can compute the gradient based on the difference between the projection of f and the measured projection. The difference can be computed by using absolute difference, squared difference, and so on.

Using gradient algorithm to iteratively find the best estimation can be defined as the following equation (eq. 2.6):

$$f^{(k+1)} = f^{(k)} - \alpha\lambda^{(k)} \tag{2.6}$$

Where $f^{(k+1)}$ is the new estimation, $f^{(k)}$ is the current estimation, $\lambda^{(k)}$ is the vector which has the same direction as the local gradient (direction that the loss will have the greatest change), α is the weight that controls the rate of the decreasing step (the length of step we walk to the lowest point). Since we want to minimize the the difference, we need to use the negative direction of the gradient $-\lambda^{(k)}$.

However, using gradient to optimize the estimation exists some problems. First one is the local minima. Most of the time, gradient usually points to the local minimum point of the function but not the global minimum. Fig 2.10 shows an example of that. If we start from point A, then the negative gradient points to the local minima point B instead of the global minima point C. Besides, the estimation might keep bouncing around the minimum point, as it cannot make sure the next estimation is exactly the minimum one. A better choice is to use conjugate gradient (CG) instead. In normal gradient method, the vector λ only considers the current gradient. While in CG, λ will consider the gradients in the previous steps as well. In practice, CG takes fewer steps to converge comparing the normal gradient method.

Maximum Likelihood Expectation Maximization

Based on the probability theory, Maximum Likelihood Expectation Maximization (MLEM) focuses on finding the best general solution of f [17]. A certain observation g is coming from a specific f. The goal of MLEM is to find the mean number of radioactive particles $\bar{f}$ in the image (or 3D object) that has the highest likelihood to produce g.

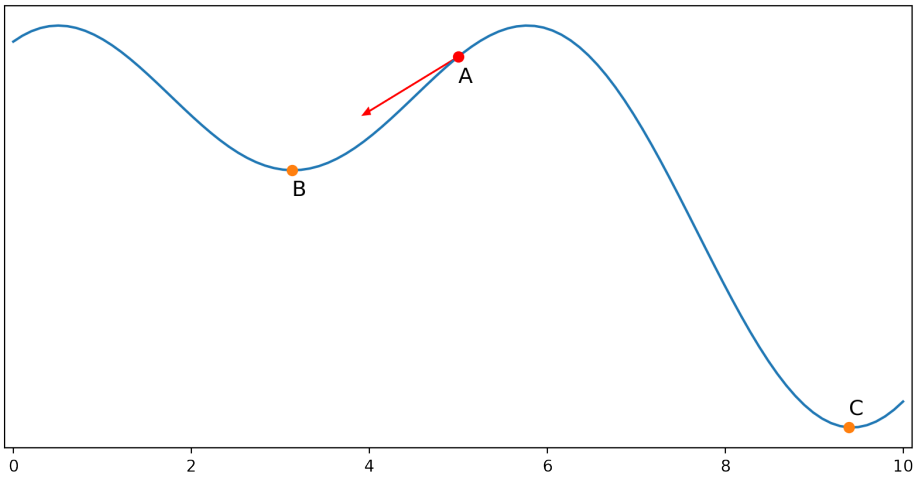


Figure 2.10: Local minima problem in gradient. If we start from A, then the gradient method tells us we should go to B. But actually C is the global minima which is what we need.

In nuclear medicine, it has been proved that the amount of the particles that the camera can detect (which is g in the formula $g = Af$) follows the Poisson distribution. Recall that the Poisson probability mass function is given by:

$$P(X = k) = \frac{e^{-\lambda}\lambda^k}{k!} \tag{2.7}$$

where k is the actual count of occurrences of event X, e is the Euler number, λ is the expectation of X and also its variance. In our case, say event X is the amount of particles obtained in bin i, then we can replace k by g_i and λ by $\bar{g}_i$, and obtain the formula of the probability of collecting g_i particles in bin i:

$$P(g_i) = \frac{e^{-\bar{g}_i}\bar{g}_i{}^{g_i}}{g_i!} \tag{2.8}$$

Note the mean of g_i is the sum of the mean count of particles emitted from all the pixels:

$$\bar{g}_i = \sum_{j=1}^{M} a_{ij}\bar{f}_j \tag{2.9}$$

We want the best estimation of $\bar{f}$ that can obtain g with the highest likelihood. In other words, we want to maximize the probability that the observation g happens given $\bar{f}$, thus $P(g|\bar{f})$. That is:

$$L(\bar{f}) = P(g|\bar{f}) = P(g_1, g_2, ...g_n|\bar{f}) = \frac{P(g_1, g_2, ...g_n, \bar{f})}{P(\bar{f})} \tag{2.10}$$

Because Poisson variables are independent, by Bayes' theorem, equation (2.10) can be written as:

$$L(\bar{f}) = \frac{P(g_1, g_2, ..., g_n, \bar{f})}{P(\bar{f})} \tag{2.11}$$

$$= \frac{P(g_1)P(g_2)...P(g_n)P(\bar{f})}{P(\bar{f})} \tag{2.12}$$

$$= P(g_1)P(g_2)...P(g_n) \tag{2.13}$$

Conventionally, log-likelihood function $l(\bar{f})$ is always used instead of using likelihood direction. Because: 1) simplify the computation (multiplication becomes addition, so that to compute the derivative is easier), and 2) they have the same monotonicity (monotonic increasing function, means the maximum point for both likelihood and log-likelihood functions are the same). Therefore:

$$l(\bar{f}) = \ln(P(g_1)P(g_2)...P(g_n)) \tag{2.14}$$

$$= \ln(P(g_1) + P(g_2) + ... + P(g_n)) \tag{2.15}$$

$$= \sum_{i=1}^{n} \ln(P(g_i)) \tag{2.16}$$

$$= \sum_{i=1}^{n} \ln \frac{e^{-\bar{g}_i} \bar{g}_i^{g_i}}{g_i!} \tag{2.17}$$

$$= \sum_{i=1}^{n} (-\bar{g}_i + g_i \ln(\bar{g}_i) - \ln(g_i!)) \tag{2.18}$$

Introducing equation (2.9) to the above one, we obtain our likelihood function:

$$l(\bar{f}) = \sum_{i=1}^{n}(-\sum_{j=1}^{M} a_{ij}\bar{f}_j + g_i \ln(\sum_{j=1}^{M} a_{ij}\bar{f}_j) - \ln(g_i!)) \tag{2.19}$$

To maximize the likelihood function, we can compute the derivative of the function and make it be zero:

$$\frac{dl(\bar{f})}{d\bar{f}} = -\sum_{i=1}^{n} a_{ij} + \sum_{i=1}^{n} \frac{g_i}{\sum_{j=1}^{M} a_{ij}\bar{f}_j} a_{ij} = 0 \tag{2.20}$$

If we multiply $\bar{f}$ on both sides, equation still holds:

$$-\bar{f}\sum_{i=1}^{n} a_{ij} + \bar{f}\sum_{i=1}^{n} \frac{g_i}{\sum_{j=1}^{M} a_{ij}\bar{f}_j} a_{ij} = 0 \tag{2.21}$$

Thus:

$$\bar{f} = \frac{\bar{f}}{\sum_{i=1}^{n} a_{ij}} \sum_{i=1}^{n} \frac{g_i}{\sum_{j=1}^{M} a_{ij}\bar{f}_j} a_{ij} \tag{2.22}$$

Now we obtain the iterative form of MLEM reconstruction method:

$$\bar{f}^{(k+1)} = \frac{\bar{f}^{(k)}}{\sum_{i=1}^{n} a_{ij}} \sum_{i=1}^{n} \frac{g_i}{\sum_{j=1}^{M} a_{ij}\bar{f}_j^{(k)}} a_{ij} \tag{2.23}$$

2.2 Artificial Neural Network

2.2.1 Fully Connected Neural Network

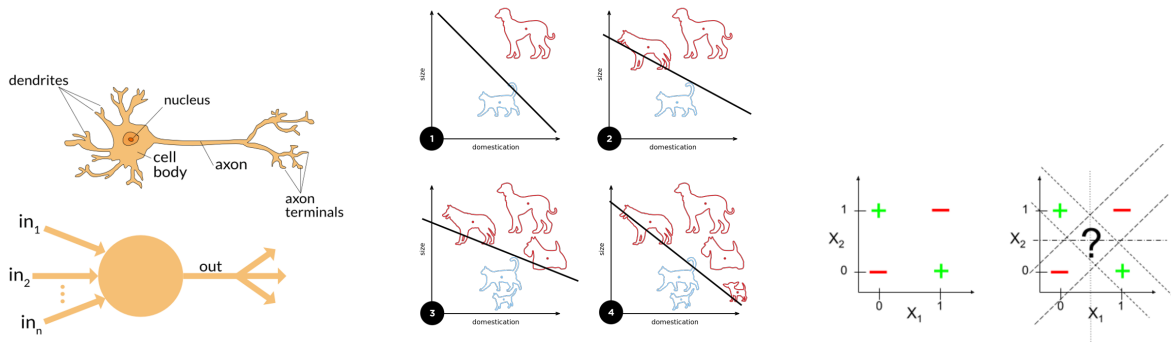
In 1943, a computational model for neural network was created [18]. In 1957, the simplest feed-forward artificial neural network was invented, called "Perceptron [19] [20]"(fig 2.11a). Perceptron is a simple abstraction of the biological neuron. The core

function of the perceptron is nothing but weighted sum (eq 2.24):

$$Output = f(\sum_{i=1}^{n} w_i x_i + w_0) \tag{2.24}$$

f is the activation function. Using that we can transfer the output from $(-\infty, +\infty)$ to "yes or no". w_0 is the bias. x_i, for $i = 1, 2, 3, ...$ are the inputs, and set $x_0 = 1$. Perceptron learning algorithm is very similar to the stochastic gradient descent:

1. Initialize $w_i = 0$, for $i = 0, 1, \cdots, n$
2. Traverse all the data until find a point such that the result of $f(\sum_{i=1}^{n} w_i x_i + w_0)$ is negative, Then update the weights: $w_i \leftarrow w_i + \alpha x_i$ for $i = 1, 2, \cdots, n$.
3. Repeat step 2 until no more mistake.



(a) (b) (c)

Figure 2.11: Perceptron: (2.11a) A biological neuron and artificial neuron [4]. (2.11b) How does a perceptron learn. Picture from Wikipedia: perceptron. (2.11c) XOR problem [5] for single perceptron.

However, it was proved that the perceptron cannot solve the non-linear problems [21]. For example, the XOR problem(fig 2.11c) is unfeasible for a single perceptron, since in the XOR problem, we cannot use a single line to divide these two categories. It is very easy to solve – use more perceptrons(fig 2.12). In fact, a naive solution is to use more layers and nodes at each layers. The first and last layers are the input and output.

Those layers between the input and output layers are called hidden layers. Every single node in hidden layers is a perceptron. This architecture is called "Multilayer Perceptron (MLP)" [22–24]. Since this structure is more sophisticated and hard to use single perceptron learning algorithm, people need to use more complex algorithm such as backpropagation [23] to make MLP learn things.

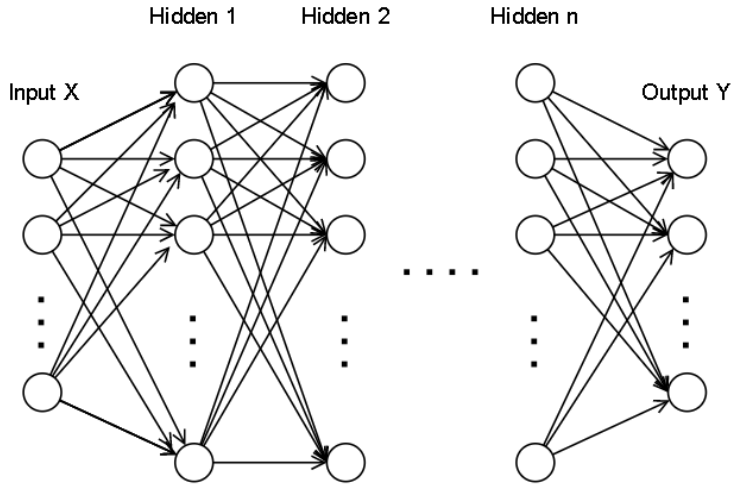


Figure 2.12: Fully connnected MLP. Sometimes called dense network. The number of nodes in the input and output layers is normally fixed. But the number of hidden layers and the hidden nodes of the hidden layers are vary.

2.2.2 Convolutional Neural Network

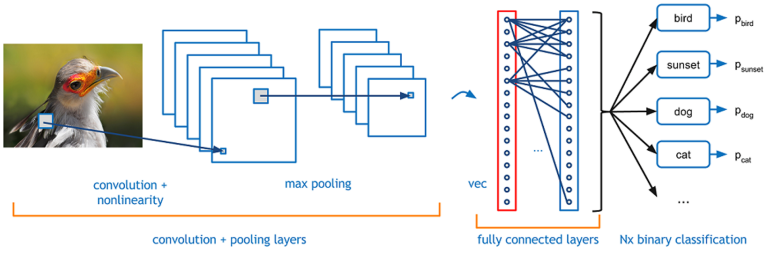
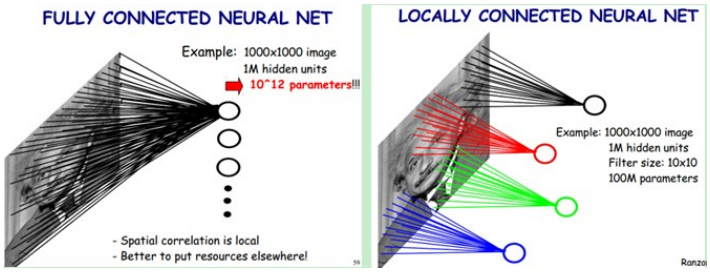


Figure 2.13: Convolutional Neural Network [6].

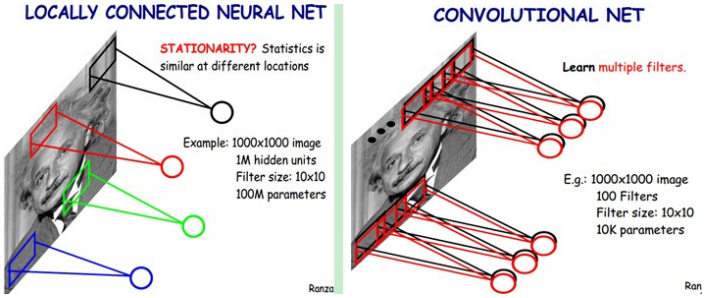
It will be very expensive to train the MLP if every layer is fully-connected to each other. For example, if each hidden layer has 10000 nodes and the input image is

1000×1000, then each layer will maintain $1000^2 \times 10000 = 10^{10}$ number of parameters. Because this network is fully connected, which means every single node connects to all the nodes in the next layer. Also, because we have more than one layer and do back propagation, in every step, the system has to update $n \times 10^{10}$ number of parameters.

Therefore, people invented a variation of MLP, convolutional neural network (CNN) [25]. Compared to the fully connected network, CNN is much more specialized and efficient. In CNN, there are two important methods used to reduce the computation load: locally connectivity(fig 2.14a) and shared weights(fig 2.14b). Local connectivity



(a) Fully connected (left) and locally connected (right) [26].



(b) Shared weights [26].

Figure 2.14: CNN.

is also called sparse connectivity. That is, instead of being fully connected, every node only connects to one part of the input, say 10, then the amount of the parameters will be $10 \times 10 \times 10,000 = 10^6$. Now we only have $\frac{1}{10^4}$ of the original number of parameters. Based on sparse connectivity, every node shares the weight. That is, share this 10×10 weights to rest of the nodes (fig 2.14b). Therefore, no matter how many nodes used in each layer, the number of parameters will be 10×10 only. In CNN, we call this 10×10

weighs "convolution kernels" or "filter". 10×10 is the kernel size or filter size. One filter will produce one feature from the input image. Normally, we will use multiple filters to get more features from the image. Using different filters, different features will be produced. This is what we called "feature map".

2.2.2.1 Transposed Convolution

CNN models got significant improvements on the image classification and recognition. But people didn't completely understand why CNN is much better than the traditional methods. So the researchers developed a technique to visualize the output of the intermediate hidden layers [27, 28]. At first, it was called "deconvolution", since it's doing upsampling which is the opposite way compared to "convolution". Now we normally call it "transposed convolution", because the term "deconvolution" is ambiguous [29–31]. There already exists a deconvolution technique. It refers to a method of removing the effect of filtering. For example, the original image (or signal) is clear, but the image observed through the cameras (or other devices) becomes blurred. If it is assumed that the function of the cameras is equivalent to a certain filter acting on the original image, which causes the image to become blurred, then according to the blurring image, the process of estimating this filter or restoring the original clear image based on the blurred image is called deconvolution.

On the other side, deconvolution in deep learning is actually doing the transposed convolution. For the feature visualization and backpropagation, there is no actual layer for the transposed convolution. It's only a process using the same filters as the convolution step (filter needs to be transposed). But in other situations like image segmentation and decoder/generator, there are transposed convolution layers, and the filters are computed by training. To understand how the transposed convolution works, let's see the process of convolution. Figure. 2.15 shows an example of the convolution

calculation given a 2×2 kernel and a 3×3 input. One should notice that in the mathematical definition, the convolution function takes the mirror of the kernel, while in neural network it uses the original order of the kernel.

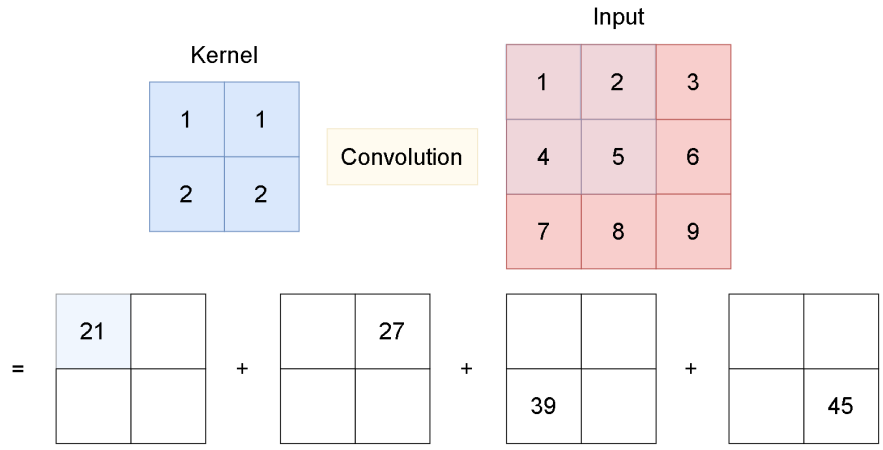


Figure 2.15: An example to show how the convolution works. Here we have a 2×2 kernel and a 3×3 input, with the unit stride and zero padding. Every pixel in the kernel will multiply the corresponding pixel in the input and then all the products will be summed up to get one pixel value in the output. The unit stride means every time the kernel will move one pixel to compute the next output pixel. Zero padding here means there is no padding for the input to expand its size.

Note that this process can be written as a matrix multiplication. The 3×3 input X can be reshaped to 1-D vector:

$$X = (1, 2, 3, 4, 5, 6, 7, 8, 9)^T$$

And do the same thing for the output reshaping:

$$Y = (y_0, y_1, y_2, y_3) = (21, 27, 39, 45)$$

Our kernel is:

$$\begin{bmatrix} k_0 & k_1 \\ k_2 & k_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}$$

To do matrix multiplication, we need to change the kernel to a specific sparse matrix K:

$$K = \begin{bmatrix} k_0 & k_1 & 0 & k_2 & k_3 & 0 & 0 & 0 & 0 \\ 0 & k_0 & k_1 & 0 & k_2 & k_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & k_0 & k_1 & 0 & k_2 & k_3 & 0 \\ 0 & 0 & 0 & 0 & k_0 & k_1 & 0 & k_2 & k_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 2 & 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 2 & 2 \end{bmatrix}$$

Now we have:

$$K \times X = \begin{bmatrix} 1 & 1 & 0 & 2 & 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 2 & 2 \end{bmatrix} \times \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{bmatrix} = \begin{bmatrix} 21 \\ 27 \\ 39 \\ 45 \end{bmatrix} = Y$$

Here we show the convolution calculation in neural network, and represent the convolution in matrix multiplication form $KX = Y$. Next we will show how the transposed convolution works and why it is called "transposed".

Typically, transposed convolution is the way to upsample the input. For example, we use the previous mentioned Y as the input here and the same kernel k. The output dimension of this process is 3×3, which is the same as the input in the convolution step. In order to compute upsample the input from 2×2 to 3×3, we take every pixel in the input and multiply with the every element in the kernel to calculate the corresponding

output pixel. Notice that there are some elements overlapped. The solution is simply to sum them up. After this upsampling step, the output is:

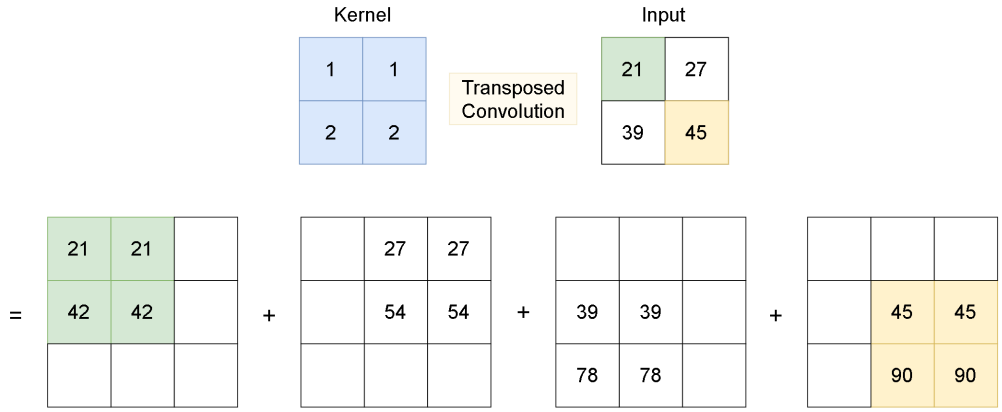


Figure 2.16: A transposed convolution example. The input here is the output in the previous convolution step, and the kernel is the same. We use different colors to display the output pixel position. For instance, when taking the green pixel of the input to multiply the kernel elements, the position of the four resulting pixels are shown using the same color in the output.

$$(21, 48, 27, 81, 180, 99, 78, 168, 90)^T$$

However, if we use the transpose of the sparse matrix of the kernel K^T multiply Y, we will obtain the same result as the previous upsampling step. That's why people called

it "transposed" convolution, because it's doing convolution with the transposed kernel:

$$K^T \times Y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 2 & 2 & 1 & 1 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 21 \\ 27 \\ 39 \\ 45 \end{bmatrix} = \begin{bmatrix} 21 \\ 48 \\ 27 \\ 81 \\ 180 \\ 99 \\ 78 \\ 168 \\ 90 \end{bmatrix}$$

Chapter 3

Related Work

This chapter is dedicated to review the related works. Firstly, we provide an overview of some typical deep learning methods that are used in image processing. Next, the applications of deep learning in medical imaging are demonstrated.

3.1 Artificial neural network

3.1.1 Convolutional neural network

CNN is one of the most important approaches in deep learning. From 1989 to 1998, LeCun et al. published a series of papers to elaborate a feed-forward neural network and how to train it [32–35]. This network is called LeNet or LeNet-5, as one of the earliest CNN. They confirmed that in handwriting character recognition, LeNet had the outstanding performance compared to all other models [35]. This simple network defines the basic elements of CNN: convolution, pooling, and dense (fully connected) layers. Because of the limited computing power, people at that time didn't pay to much attention on it. In the 21st century, researchers started to use graphics processing

unit (GPU) to accelerate the CNN computation and won some image competitions [36–39]. In 2012, AlexNet won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) and its top-5 error rate was 10% more lower than other the runner up [40,41]. AlexNet has an significantly influence in computer vision, stimulating many studies on CNN and GPU acceleration. Now, CNN is widely used in image recognition [42–46], natural language processing (NLP) [47–49], and other areas like medicine [50, 51] and checker game [52–54].

3.1.2 U-net

U-net is one of the expansions of CNN. Olaf et al. proposed this special architecture for biomedical image segmentation [55]. It is an improvement of fully convolutional network (FCN) [56]. An ordinary U-net structure contains two symmetric components: a contracting path and a expanding path. The contracting path is a common CNN. On the other hand, the expanding path is similar to the contracting path, except concatenating the features from the contracting path at the same level, and using unpooling or transposed convolution to do upsampling. Most of the time, U-net was applied in biomedical image segmentation, like Brain Tumor Segmentation (BraTS) [57–59], Segmentation of the Liver Competition 2007 (SLIVER07) [60–62]. There are some other applications of U-net. Çiçek et al. demonstrated their 3D U-net for volumetric segmentation which learned from sparsely annotated volumetric images and achieved good performance of a complex, highly variable 3D structure, the Xenopus kidney. Nazem's group used a improved 3D U-net to predict the binding sites of the proteins and therefore, helped the drug design for the novel proteins [63]. Vladimir Iglovikov and Alexey Shvets showed that using a pre-trained encoder (the contracting path) can improve the performance of U-net in image segmentation and won the Kaggle competition of Carvana Image Masking Challenge [64].

3.1.3 Attention Mechanism

The *attention* mechanism has been applied in many areas. Originally it was applied in computer vision and then rapidly for the learning long-range association between data [65–67], like the semantics of a word in a sentence may be determined by the context in the sentence at many words apart. Due to the presence of long-range data association in input data, it is important to pay attention to a particular region of the data and reveal the connection between two areas which are not close to each other. When we look at some information (i.e sentences, image, ...), we pay more attention to those aspects of data, which are perceived to be more important. Initially, the attention mechanism appeared in computer vision, and rapidly grew in deep learning, especially natural language processing (NLP) [65] [66]. Now, areas such as computer vision and pattern recognition have begun to use attention to produce better results. Wang, et al. [67] presented non-local operations, which originated from the non-local means method, to "compute the response at a position as a weighted sum of the features at all positions." Zhang, et al. modified Wang's work and used it in the generative adversarial network (GAN), called self-attention GAN (SAGAN) [68]. SAGAN can generate the image by using cues from features across different locations. In this paper, the author claimed they obtained better performance on the challenging *ImageNet* dataset. In other words, the attention component can be embedded into a convolutional network to enhance the reconstruction quality, as observed in our experiments with the proposed ANN.

3.2 Deep learning in medical imaging

Recently, due to the rapid development of deep learning (DL) techniques, increasing research is being devoted towards the potential of using an ANN model to improve

image reconstruction. A survey from McCann et al. reviewed the recent applications of CNNs to the inverse problems such as denoising, deconvolution, super-resolution, and medical image reconstruction [69]. Improvements are shown over traditional techniques, including sparsity-based techniques, e.g., compressed sensing. The authors attempted to address some important questions in the application of deep learning in inverse problems, such as, where do the training data come from; what are the impacts of architectures of the CNN; and, how is the learning problem formulated and solved? There are other deep learning applications dedicated to medical imaging techniques [70]. They can be used for image reconstruction, image conversion (from one modality to another), pre- and post-processing, etc.

3.2.1 Image denoising

A specific application of deep learning in medical imaging is image denoising. Wang, et al. [71] proposed an approach that is useful in recovering wavefronts from direct intensity measurements, imaging objects from diffusely reflected images (as in ultrasound or seismic imaging), and denoising the scanning transmission electron microscopy images. They claimed its utility in solving arbitrary inverse problems. Instead of estimating source data directly from observations, Zou, et al. [72] trained a deep neural network to estimate the *degradation parameters* under an adversarial training paradigm, and applied their method to a variety of real-world problems including image denoising, image de-raining (raining being a type of noise commonly found in old television sets), shadow removal, non-uniform illumination correction, and under-determined blind source separation of images or speech signals. The *AUTOMAP* [73] initially used two fully connected (FC) layers to learn the linear mapping, and then used several convolutional layers over the output of the FC layers for denoising.

Heinrich, et al.'s model [74] combined a fully convolutional network (FCN) and a U-

net using residual connections for low-dose CT image denoising and obtained promising results on the XCAT phantom data. Reymann, et al. [75] exploited a four-layer U-net and proved this model can significantly increase image quality when using the data obtained from Monte Carlo simulations of XCAT phantoms. In Liu, et al.'s paper [76], a couple U-net structures (CU-net) was considered to improve the detectability of perfusion defects compared to the traditional Gaussian post-filter for denoising. The human data result demonstrated that CU-net had a stronger capability for noise suppression and perfusion defect detection as compared to the traditional Gaussian filtering. Po-Yu Liu and Edmund Y. Lam [77] created a deep learning structure to do the Poisson image denoising, and got a significant improvement than the traditional denoising algorithms. Gong K, et al [78] introduced a modified U-net [55] for PET image reconstruction, and got a better result than the Gaussian filter and anatomically-guided reconstructions using the kernel method or the neural network penalty.

3.2.2 Motion correction

As previously mentioned, there exists a multitude of literature and algorithms on motion correction in many modalities, primarily in the context of cardiac imaging as well as imaging of other organs affected by motion. Following is a select list of relevant works. Dubbs, et al. [79] introduced a Fourier-transform approach to accelerate the computation of the L_2 norms and the FFT convolutions for tMC which was the first step in a pipeline of algorithms to analyze calcium imaging videos. Besides MC in calcium imaging, Min, et al. [80] proposed a coronary MC algorithm in computed tomography (CT) angiography. In Single Photon Emission Computed Tomography (SPECT), an MC algorithm called *MoCo* from Cedars-Sinai is licensed by most vendors in cardiac SPECT and embedded in their iterative reconstruction algorithms [81,82]. Rather than correcting the organ motion, Mitra, et al. [83] provided a tool *SinoCor* that can detect

and correct the patient motion on camera-projection data for SPECT. The deep partial angle-based motion compensation (Deep PAMoCo) [84] made use of a CNN to predict the motion model to aptly determine the motion vector field (MVF). It has been shown that the Deep PAMoCo provided an efficient approach for the improvement of image quality and processing time. U-net was also used for MC as a post-processing step after image reconstruction [85, 86].

3.2.3 Image Reconstruction

Zhang and Zuo [87] proposed a method using a recurrent neural network (RNN) for computed tomography (CT) image reconstruction such that the reconstruction with the total variation (TV) prior was significantly improved, especially for those images containing small metal objects. Fu and De Man [88] decomposed the reconstruction problem into hierarchical sub problems, and designed a neural network of six hierarchical stages to solve the sub-reconstruction problems, level-by-level. Li et al. [89] developed an intelligent CT network (iCT-Net), that can reconstruct images with high quantitative accuracy with either complete or incomplete line integral data. Wu et al. [90] used a k-sparse autoencoder [91] to learn the nonlinear sparse prior from normal-dose CT images reconstructed by FBP, and then applied this model to the iterative reconstruction of the low-dose data. Lim et al. [92] proposed the BCD-Net to be the regularization method for the low-count PET reconstruction, and showed significant improvements compared to non-trained regularizers, total variation (TV) and non-local means (NLM). Ouyang et al. [93] used a generative adversarial network (GAN) to reconstruct high-quality and accurately pathological features of standard-dose amyloid PET images from ultra-low-dose PET images. Häggström et al. [94] presented a deep convolutional encoder-decoder network that can reconstruct the images directly from the PET sinograms. Our proposed approach is based on Häggström's network model.

Some methods unified iterative reconstruction with deep learning, like EM-Net [95] and MAPEM-Net [96]. The original maximum a posteriori probability expectation maximization (MAP-EM) algorithm [97] is an extension of the expectation maximization method for maximum likelihood image reconstruction in emission tomography (MLEM) [17, 98]. Both MLEM and MAP-EM can reconstruct the image by several iterative steps. EM-Net replaces the penalty item in the original (MAP-EM) algorithm by a modified U-net [55] (developed for image transformation), while the MAPEM-Net are combined with the iterative reconstruction more deeply. In the MAPEM-Net, there are nine modules, and each module performs two MAPEM steps (maximum likelihood estimation and expectation maximization), followed by an independent U-net.

Chapter 4

Preliminary Experiments

In this chapter, we show some of our preliminary work using deep learning to solve problems, including the parameter prediction, inverse problem, and motion correction. All the data we used in these works is created by ourselves, in order to explore the ability of the neural network in the area of the image transformation.

4.1 Parameter Prediction

4.1.1 Fourier Transformation

Fourier transform (FT) is widely used in medical imaging for image analysis and processing. FT is a mathematical technique that decomposes a signal or image into its frequency components. In medical imaging, it is used to transform signals from the time domain into the frequency domain, making it possible to analyze and manipulate the frequency content of the image. In medical reconstruction, FT provides a powerful tool as it allows the extraction of useful information from acquired data (like sinogram) by transforming it into the frequency domain, filtering it, and then reconstructing it

back into the time or space domain. This technique has revolutionized medical imaging, enabling the creation of detailed and accurate images of internal structures and functions of the body.

In practice, we used discrete Fourier transform (DFT) since the data was obtained in discrete time. Most of the time, The Fast Fourier transform (FFT) algorithm is preferred over the DFT algorithm because it is computationally efficient (from $O(n^2)$ to $O(n \log n)$), requires less memory, is easier to implement, and is as accurate as the DFT algorithm for most applications. FFT was derived in 1805, but not widely used until 1965 [99]. The American mathematician Gilbert Strang believed that FFT is "the most important numerical algorithm of our lifetime" [100] [101]. Also, IEEE put FFT into the list of the top 10 algorithms of 20th century in the IEEE journey "Computing in Science & Engineering" [102].

In the following context, I show how to use a fully connected network to simulate inverse discrete Fourier transform. That is, according to the frequency-domain samples, reconstruct the certain signals in time-domain. So the frequency-domain samples will be the input, and the time-domain samples will be the output.

The training data I used is generated by the trigonometric functions, sine and cosine. Totally, there are four types of signals:

$$A\sin(2\pi \times t) + B\sin(2\pi \times 5t) + 2sin(\omega \times 3t) \tag{4.1}$$

$$A\sin(2\pi \times t) - B\sin(2\pi \times 5t) + 2sin(\omega \times 3t) \tag{4.2}$$

$$A\cos(2\pi \times t) + B\sin(2\pi \times 5t) + 2sin(\omega \times 3t) \tag{4.3}$$

$$A\cos(2\pi \times t) - B\sin(2\pi \times 5t) + 2sin(\omega \times 3t) \tag{4.4}$$

While t is the time, whose range is [0, 5), step 0.005. A and B have the same range: [1, 3), step = 0.2. ω is from 2π to 12π (exclusive), step π. Thus, there are 10 different A values, 10 different B values and 10 different ω values. For each type, there are $10 \times 10 \times 10 = 1,000$ different signals. Totally, I have $4 \times 1,000 = 4,000$ signals. Use different amplitudes (A and B), and angular frequency ω to create different signals. Do FFT on these signals, and the result of that will be the input of my neural network. The original signals will be the outputs (or the labels).

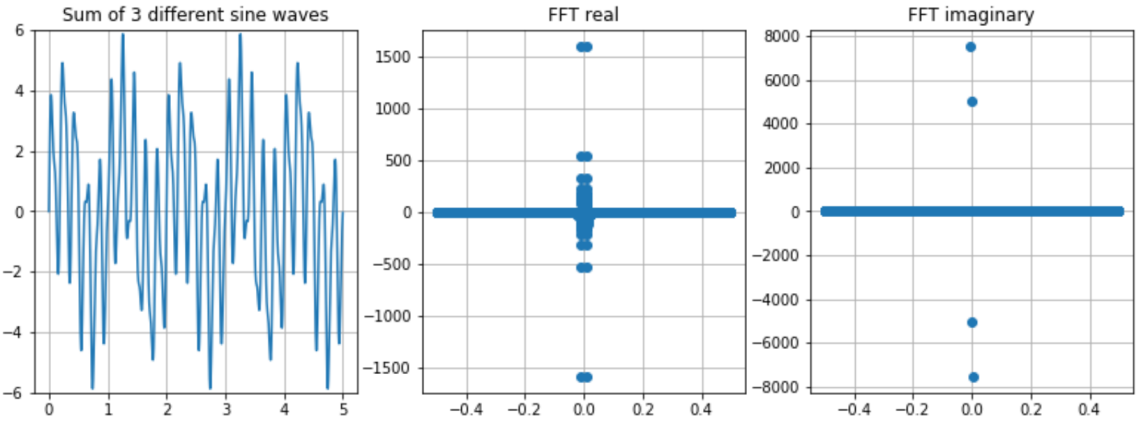


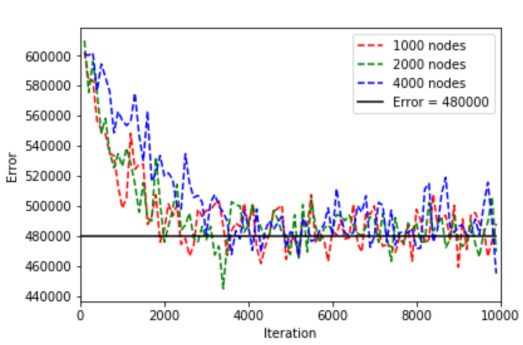
Figure 4.1: One of the signals and the corresponding FFT (real and imaginary). The signal function is: $2\sin(2\pi \times t) + 3\sin(2\pi \times 5t) + 2sin(7\pi \times 3t)$

My neural network model is a fully connected network (FCN). Totally, I have 4,000 signals, then I randomly chose 100 of them to be the testing set and rest of them to be the training set. The number of the training iterations is 10,000. In every iteration, randomly choose 100 signals from the training set to train the network. Loss function is the sum of the squared difference(eq 4.5).

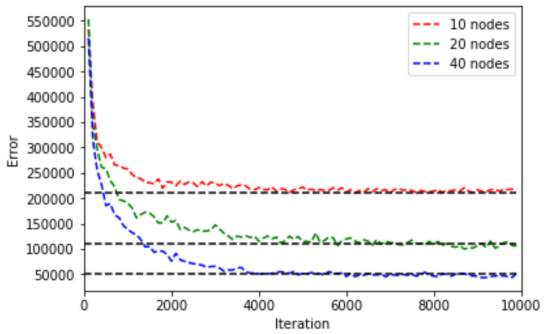
$$error = \sum(\vec{x_i} - \vec{x_i}')^2 \tag{4.5}$$

where $\vec{x_i}$ is the original 1D signal, $\vec{x_i}'$ is the reconstructed signal.

Actually, using too many hidden nodes per hidden layer will not improve the result. fig 4.2 shows the error curve during training. All of them use 4 hidden layers, but different number nodes. In both fig 4.2a and fig 4.2b, Y-axis is the error value, computed by eq 4.5. X-axis is the number of iterations. In every iteration, randomly choose 100 signals from the training set to train the network. And the reconstructed signal is in fig 4.3.



(a) Using lots of nodes per hidden layer.



(b) Using fewer nodes per hidden layer.

Figure 4.2: Using 4 hidden layers but different number of hidden nodes.

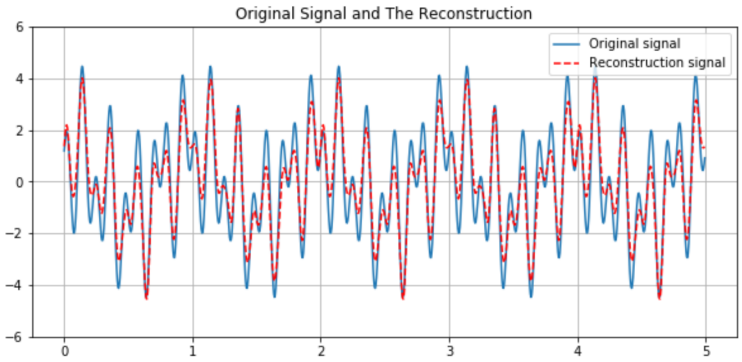


Figure 4.3: Original signal and the corresponding reconstruction. RMS is 0.6890. X-axis represents time, while y-axis is the signal intensity.

4.1.2 Attenuated Uniform Disk

Real data may not be "real" since it will be corrupted by attenuation. When a beam penetrates a volume of any material, photons will be absorbed or scattered. Therefore, the detectors will lose counts. Attenuation is one of the main problem in medical imaging. It may increase noise, or distort the image. One of the ways to correct for attenuation is Chang's method [103]. Each pixel of the object is multiplied by a coefficient of attenuation (eq 4.6).

$$C(x,y) = (\frac{1}{M}\sum_{i=0}^{M} e^{-\mu l_{\theta_i}})^{-1} \tag{4.6}$$

where M is the number of projections. l_{θ_i} is the distance from (x,y) to the boundary of the object (see fig 4.4) In order to "correct" the image, the additional information

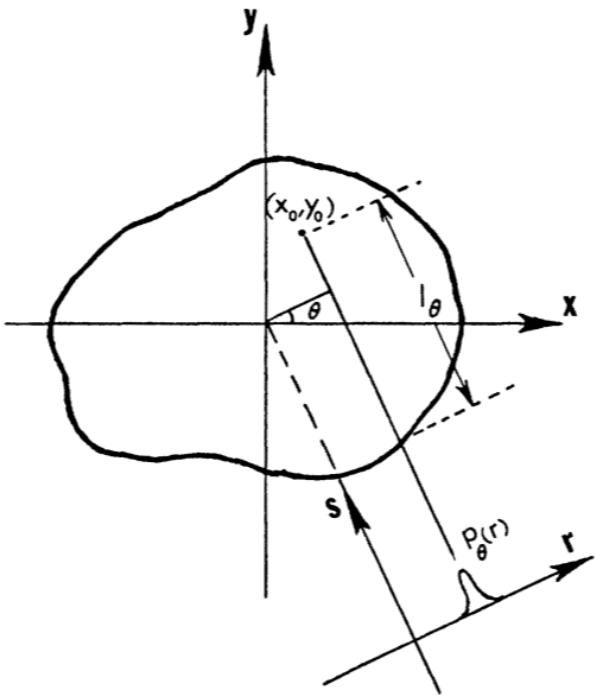


Figure 4.4: Illustrate the parameters used in formula of the attenuation correction coefficient in a coordinate system.

is needed when reconstructing the image. It can come from statistic features [104], or from somewhere else. For example, in medical imaging, people will use x-rays to construct the attenuation map for the correction [105, 106].

My experiment is to test if NN can automatically embed the attenuation correction while trying to reconstruction the image. I create many disks with different positions and radius. And then, use an analytical expression to compute the attenuated projections of a certain disk with a fixed attenuation coefficient. Then we build a simple convolutional neural network, use attenuated projections to be the input, to predict the position and the radius of the disk, and the attenuation coefficient.

Following equation (eq 4.7) is the analytical expression to compute the attenuated projections.

$$p(\xi) = \frac{C - Ce^{-2\mu\sqrt{R^2-\xi^2}}}{\mu} \tag{4.7}$$

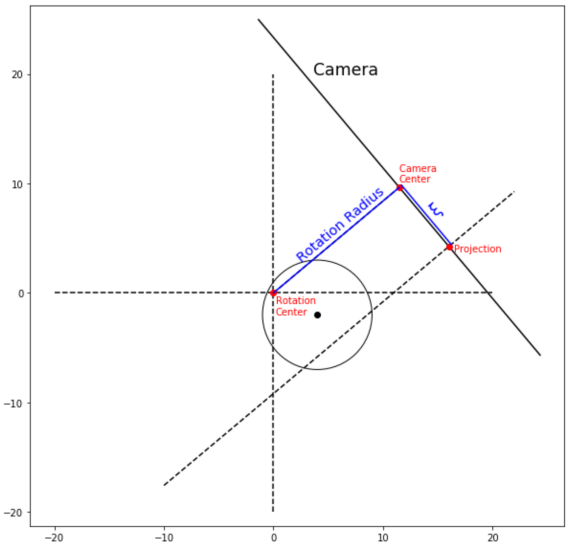
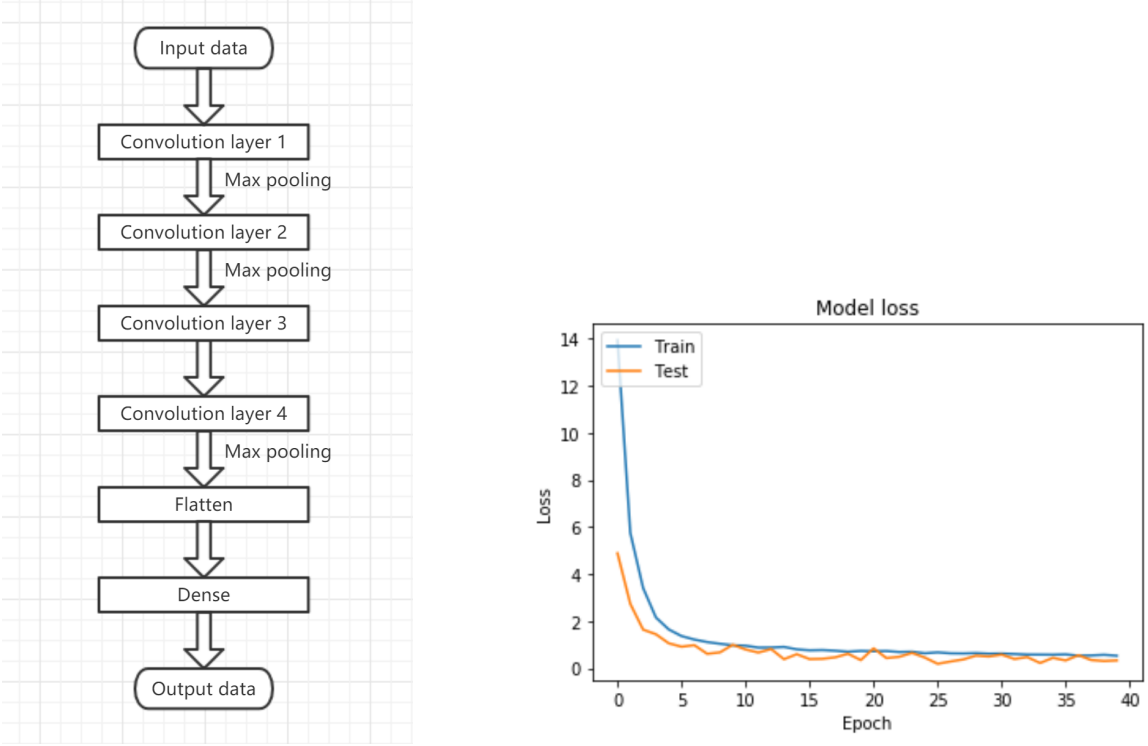


Figure 4.5: Illustrate the geographical meaning of the parameters used in analytical expression.

In eq 4.7 μ is the attenuation coefficient, C is the pixel value of the disk which is always one here, R is the radius of the disk, e is a mathematical constant, the Euler's

number (the base of the natural logarithm), ξ is the distance starting from the center of the camera ending with the projection of a disk pixel on the camera (fig 4.5).

The neural network is a convolutional neural network (fig 4.6a). In convolution layer 1 and 2, there are 16 filters. The kernel size is 5 x 5. The activation function is Rectified Linear Unit (ReLU) [107]. In convolution layer 3 and 4, there are 32 filters. The kernel size is 5 x 5. The activation function is relu. The optimizer function is Adam, learning rate 0.0001. Loss function is mean square. Batch size 30, 40 epochs.



(a) CNN model.

(b) Loss value during training.

Figure 4.6: CNN model and error curves for attenuated disk projectio

In total, there were 1764 sinograms created. 1500 of them were used for training, and rest of the 264 sinograms for validating. Fig 4.6b shows the loss value during the training. Fig 4.7 shows the part of the results. Note the attenuation coefficients μ are small fraction numbers (much smaller than one) and other values are integers. Considering the accuracy, I used the exponent of μ, because this model may not give

	Prediction from CNN				Real disk			
	radius	pos_x	pos_y	μ	radius	pos_x	pos_y	μ
disk 0	4.06	8.98	-8.58	1.81E-04	5.00	10.00	-10.00	1.00E-04
disk 1	1.77	5.49	-0.87	4.22E-04	2.00	6.00	-1.00	1.00E-04
disk 2	3.32	-1.68	-9.07	3.14E-04	4.00	-2.00	-10.00	1.00E-04
disk 3	2.64	-3.29	-0.89	5.44E-04	3.00	-4.00	-1.00	1.00E-04
disk 4	2.65	6.35	-3.61	4.90E-04	3.00	7.00	-4.00	1.00E-04
disk 5	4.31	2.77	-3.39	3.51E-04	5.00	3.00	-4.00	1.00E-04
disk 6	1.94	8.49	-8.65	3.05E-04	2.00	10.00	-10.00	1.00E-04
disk 7	1.75	7.72	5.93	3.47E-04	2.00	8.00	6.00	1.00E-04
disk 8	1.85	-0.02	4.86	3.17E-04	2.00	0.00	5.00	1.00E-04
disk 9	1.68	-4.69	-7.51	3.93E-04	2.00	-5.00	-8.00	1.00E-04
disk 10	4.35	7.98	-4.44	4.17E-04	5.00	9.00	-5.00	1.00E-04
disk 11	4.32	-6.05	1.73	5.65E-04	5.00	-7.00	2.00	1.00E-04
disk 12	3.50	9.14	7.38	2.40E-04	4.00	10.00	8.00	1.00E-04
disk 13	2.04	-3.85	8.62	1.75E-04	2.00	-4.00	9.00	1.00E-04
disk 14	1.64	-4.41	-4.67	4.84E-04	2.00	-5.00	-5.00	1.00E-04
disk 15	2.54	-6.29	-3.85	4.98E-04	3.00	-7.00	-4.00	1.00E-04
disk 16	3.52	-9.36	7.33	1.93E-04	4.00	-10.00	8.00	1.00E-04
disk 17	3.53	6.32	-6.26	4.64E-04	4.00	7.00	-7.00	1.00E-04
disk 18	4.42	6.07	1.67	5.13E-04	5.00	7.00	2.00	1.00E-04
disk 19	3.48	-3.20	2.46	5.70E-04	4.00	-4.00	3.00	1.00E-04

Figure 4.7: Some results. You can see there are two parts in this big table. The left one contains the results from the CNN. The data from the right one is the real data that is used to generate the sinograms.

me a good result for μ if directly using it for training. For example, if μ is 0.00001 and the output of μ is 0.001, the error is still very low because other numbers are much larger than this μ value, but actually the predicted μ value is 100 times more than the actual μ value. That's why instead of directly using 10^{-k}, I used k to train the network. In 4.7, I converted back to the actual μ values in order to understand easily.

4.2 Synthetic Object Reconstruction

In order to simulate the myocardium, we chose U-shape object to be the ideal image. The size of all the generated images is 64x64. There are four basic U-shapes which are shown in fig 4.9. According to those basic shapes, translation and flipping are applied on them to create a large data set. Algorithm shows in fig 4.8. There are 640 for each shape. In total, 2560 images were created. In each shape of images, 40 images are randomly selected to be the testing set, and rest of the 600 images to be

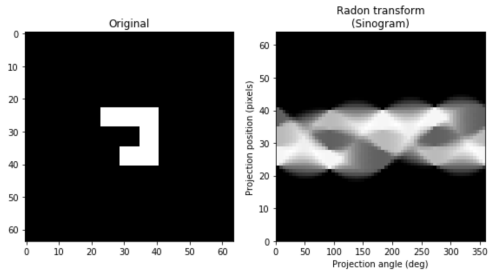
the training set. Thus, the testing set has $4 \times 40 = 160$ images, and the training set has $4times(640 - 40) = 2400$ images.

```
def generate_more(object_img):
    # store generated images
    imageSet = []    # for images
    sinogramSet = [] # for sinograms

    # x and y for image translation (shifting)
    for x in range(-16, 16, step = 2):
        for y in range(-10, 10, step = 2):
            # translation image
            temp_trans_img = translating(object_img, x, y)
            imageSet.append(temp_trans_img) # add to imageSet
            # compute the sinogram by radon transformation
            temp_sino = radon(temp_trans_img)
            sinogramSet.append(temp_sino) # add to sinogramSet

            # flip horizontal
            temp_flip1_img = flipping(temp_trans_img)
            imageSet.append(temp_flip1_img)
            # compute the corresponding sinogram
            temp_sino = radon(temp_flip1_img)
            sinogramSet.append(temp_sino)
            # end for loop
    # return
    return imageSet, sinogramSet
```

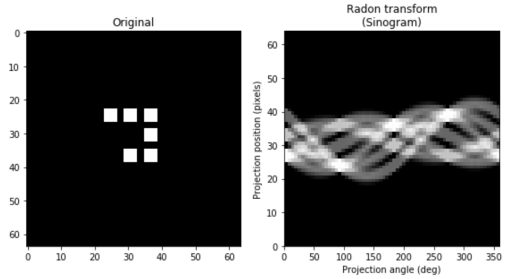
Figure 4.8: How to generate more data based on those four basic images(fig 4.9)

The neural network model I used is a FCN. There are five hidden layers in the model, each hidden layer has 100 neurons. Activation function is rectified linear unit (reLU [107] [108]), since negative value is meaningless in an image. Error function is mean squared error(MSE), also called mean squared deviation (MSD). The optimizer is Adam, because some researchers believe Adam might be the best overall choice [109]. Also, in Stanford course "CS231n: Convolutional Neural Networks for Visual Recognition", Adam is "currently recommended as default algorithm". The learning rate is 0.001

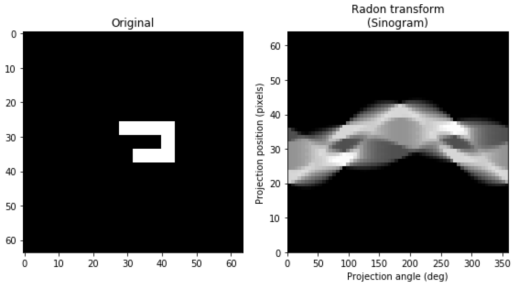
Recall that my training set has 2400 images and my testing set has 160 images.



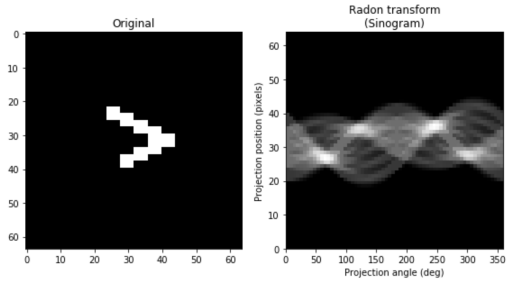
(a) Unsymmetrical U-shape image and the corresponding radon transform. Generated by several squares, each square is constructed by 6x6 pixels.



(b) Disconnected U-shape image and the corresponding radon transform. Generated by several squares, each square is constructed by 6x6 pixels.



(c) Thinner U-shape object. Generated by several squares. Square size is 4x4 pixels.



(d) V-shape object. Generated by several squares. Square size is 4x4 pixels.

Figure 4.9: Four basic shapes. Image size is 64×64.

While training, the number of epochs I set is 100, batch size 50. In each epoch, I used testing set to validate the error. The training and validation error curves are in fig 4.10. Some reconstruction shown in fig 4.11.

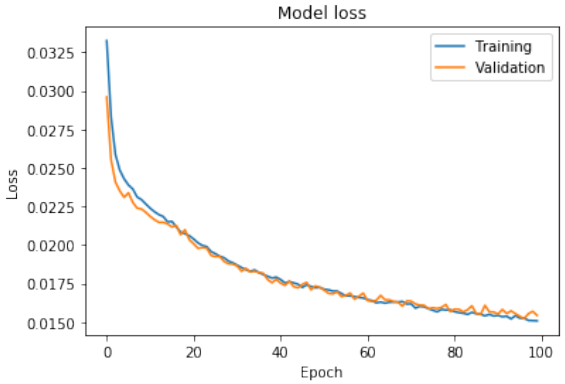


Figure 4.10: Training and validation error while training

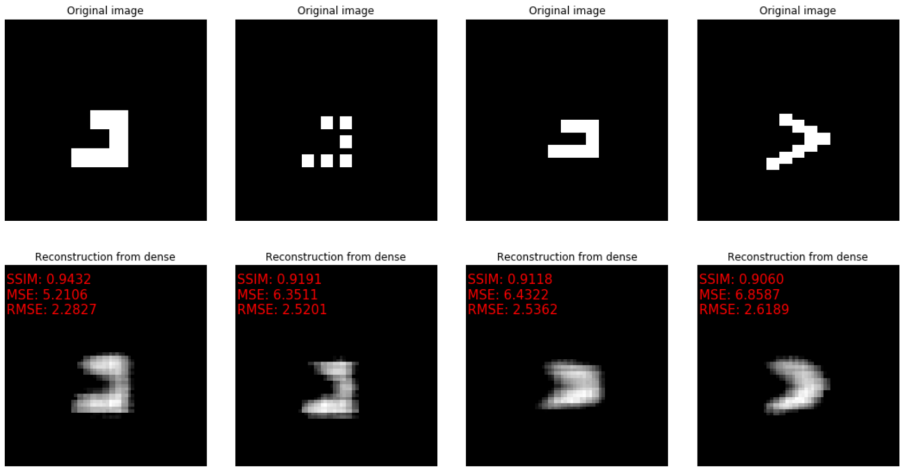


Figure 4.11: Reconstructions using FCN. The first row is the real image. The second row is the reconstruction from Dense NN. MSE, RMSE, SSIM are used to measure the difference between the reconstruction and the real image.

4.3 Motion Blur Elimination

In this work, we address motions like cardiac motion as symmetric Gaussian blur and try to recover that directly from the sinogram data. For simplicity of fast experimentation, we use 2D synthetic data that may be easily extended to 3D. Rather, we used different shapes (motivated by that of the heart) and locations of the target object to prove the robustness of our results. We train a convolutional neural network (CNN) for recovering the ground truth motion model (Gaussian function).

Four basic 2D U-shape masks and annular elliptical rings were selected to be the ideal shapes in this work (fig 4.13 and 4.12). The four basic U-shape images were augmented by changing the positions (translation) and orientations (rotation) of these images producing 2560 different images. For the different annular elliptical rings generation, two different ellipse functions were utilized: one for the outer ellipse and another for the inner one. By changing the center, the width, and the height, we got 640 different annular elliptical rings. Then we did the same augmentation operations as before for this 640 images that is included for training and validation along with the previous

abstract shapes (fig 4.12).

Figure 4.12: Annular elliptical rings. They are generated by two ellipse functions. The outer ellipse is fixed. By changing the center, the width and the height of the inner ellipse, we get different images to simulate hearts.

The symmetric Gaussian blur was decided to simulate the cardiac motion in this work. We used 8 different 2D 5×5 Gaussian filters to blur these images. The discrete Radon transform was applied to those blurred images to produce blurred sinograms. Subsequently, Poisson noise is added to the sinograms. Fig 4.14 shows this process.

Figure 4.13: Four basic shapes.

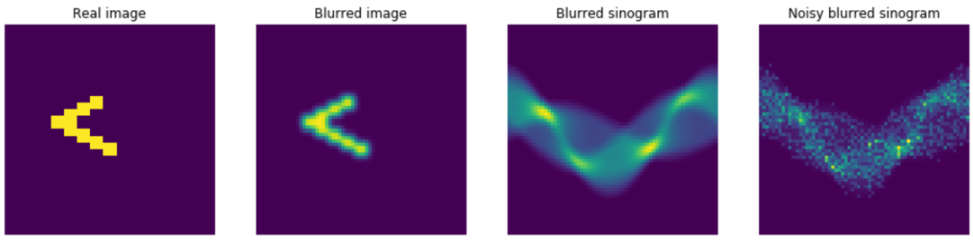


Figure 4.14: Steps to create a noisy blurred sinogram. (From left to right) Based on the real image, we used Gaussian filter to blur it to get a blurred image. Then, Radon transform this blurred image to create the blurred sinogram. Finally, by adding Poisson noise we get the noisy blurred sinogram that is used as input.

In summary, say f is the Gaussian filter, g is the image, s is the sinogram, R is the Radon transform, and $*$ is the notation for the convolution operation, then the

equation we used to generate the sinogram s is:

$$s = R(f * g) \tag{4.8}$$

Our problem is: given an s, can we recover f and g by training a deep learning model. We developed and trained two independent neural networks to: 1) learn to recover the filter f from a motion-blurred sinogram; and 2) learn to reconstruct the noise free image g from the motion-blurred sinogram. Therefore, we used two individual NNs to solve these two problems: 1) A CNN to extract the Gaussian filter from the noisy sinogram, and 2) A CED to reconstruct the image from the noisy sinogram.

4.3.1 Motion function recovery

A convolutional neural network (CNN) was used to recover the motion function. It was trained to extract the filter from the blurred sinogram. It contains three convolutional layers and two dense layers (fig 4.15). The number of the epoch is 20, batch size 50. The optimization function is Adam [110], and the loss function is the mean squared error. We used mean squared error to measure the model loss. After 100 epochs,

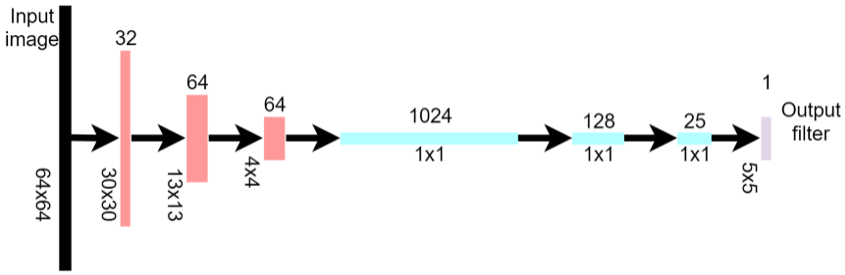


Figure 4.15: Convolutional neural network. It is used to extract the filter from the sinogram.

the training loss is 1.4249e-6 (RMS = 1.1937e-3), and the validation loss is 1.6612e-5 (RMS = 4.0758e-3). Each epoch of training takes approximately 3 sec on our machine (CPU is Intel(R) Core(TM) i7-9700K @ 3.60GHz. GPU is Nvidia Titan Xp. RAM

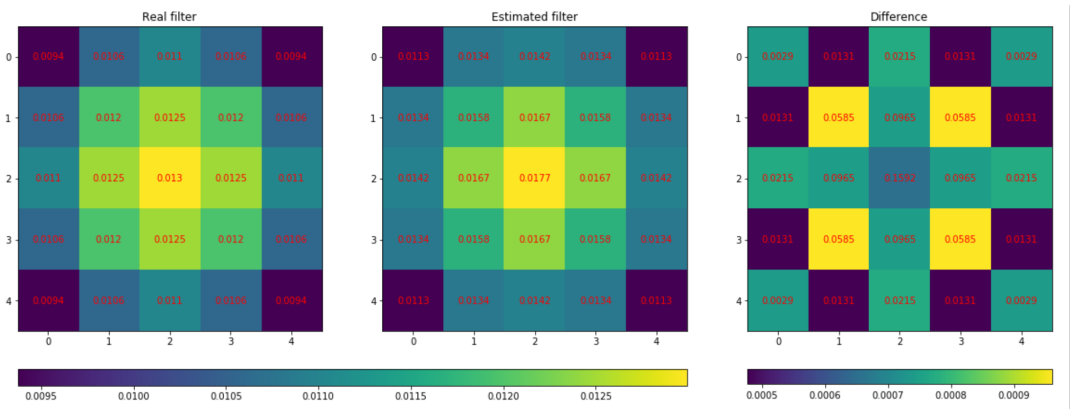


Figure 4.16: A sample result. Compare the real filter and the estimated filter. The third one shows the absolute difference between the real filter and the estimated filter.

is 32 GB). Fig 4.16 shows one sample result of the estimated filter (from the model output) compared to the ground truth filter.

4.3.2 Image reconstruction from noisy blurred sinogram

Using the same data, we used CED (fig 6.5) to reconstruct the motion-free from the noisy sinogram. We trained the network on a GPU node on a cluster with the dataset which contains the ellipse rings. The number of the epoch is 100. Batch size is 50, optimization function is Adam, and the loss function is MSE. The configuration of this GPU node is: 2 x 10 core Intel Xeon @ 2.30GHz, 131GB of RAM, 4 x Nvidia Tesla K40m. After 100 epochs, the training loss is 4.0035e-04 (RMS = 0.02001), and the testing loss is 1.928e-4 (RMS = 0.01825). Training time is 24302.638 seconds, and the average reconstruction time per image is 2.710e-3 seconds.

According to our results, we believed the neural networks have a great potential to recover motion model and reconstruct images from motion-blurred sinograms. It can improve the reconstruction process in nuclear imaging such as PET, SPECT, and CT, for example, for noise reduction. It also provides a fast way to reconstruct the image in place of iterative reconstruction methods, though it takes much more time to train

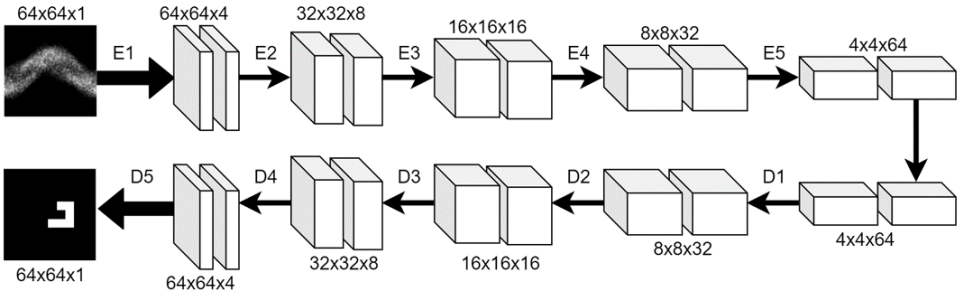


Figure 4.17: CED to reconstruct the image from a noisy blurred sinogram.

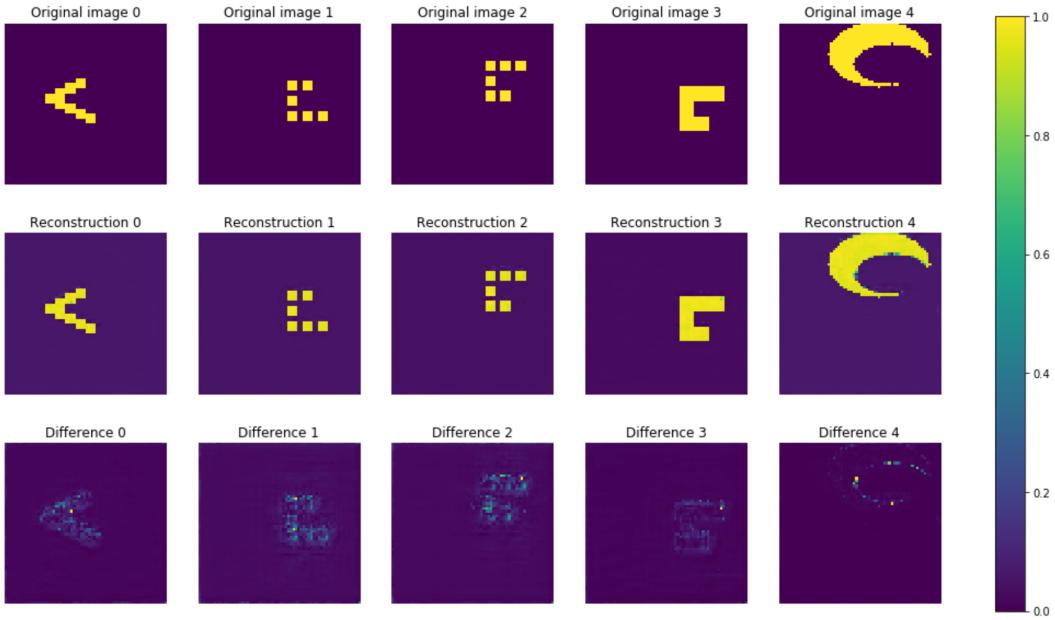


Figure 4.18: Image reconstruction with adapted CED. Annular elliptical ring on the last row simulates shapes of hearts. First row shows the ground truths. Second row shows the reconstructions. Third row is the difference between the ground truth images and the reconstruction images.

the network.

In order to avoid motion generated noise, respiratory and cardiac gating methodologies have been developed. They are quite useful for static imaging protocols with radiotracers and contrast agents, where tracer agent-concentrations are allowed to stabilize in body and imaging takes place after some wait time. However, in dynamic imaging protocols, which provide better quantitative and diagnostic information [111], and where imaging starts immediately after injection to study the pharmacokinetics of the imaging agent, gating techniques are not as effective as in static imaging. Our proposed technique will be very useful in dynamic imaging as the trained neural network will incorporate the motion model. In the near future we want to develop such motion corrected dynamic image reconstruction deep learning models.

Finally, our technique needs to be validated with real data. Availability of large real training dataset is a challenge. We believe, combination of synthetic data and limited amount of available patient data will provide quality training. Another challenge is the adaptability of a model for individualized patient. Motion model's topological shape is expected to be very similar for all patients, however, our motion model is too simple to simulate the patient motion. In the future, using real data or an enough complex simulation data as well as the motion model is our research direction.

Chapter 5

Reconstruction From Motion Blurred Sinogram Using Deep Learning

This work addresses Inverse Radon Transform (IRT) with Artificial Neural Network (ANN) or Deep Learning, while performing motion correction. The purported application domain is cardiac image reconstruction in emission or transmission tomography where IRT is relevant. Our main contribution is in proposing an ANN architecture that is particularly suitable for this purpose. We validate our approach with two types of datasets. First, we use an abstract object that looks like a heart to simulate motion-blurred Radon Transform (RT). With the known ground truth in hand, we train our proposed ANN architecture and validate its effectiveness in Motion Correction (MC). Second, we used human cardiac gated datasets for training and validation of our approach. Gating mechanism time-bins data using electro-cardiogram (ECG) signals for cardiac motion correction. We have shown that the trained ANNs can perform bet-

ter motion-corrected image reconstruction than most commonly used reconstruction techniques. We have compared our model against two existing ANN-based approaches to prove the former's superiority. In this chapter, we provide an approach for reconstructing motion-corrected image that eliminates the need for any hardware gating in medical imaging.

5.1 Introduction

The Radon transformation (RT), or the integral projection of a probed object, represents the abstract view of tomographic imaging in medicine, astronomy, and other areas. Integral projection is the signal received by an 1D- or 2D-camera pixel from the integration over 2D- or 3D-pixel values along a line-of-sight perpendicular to the camera pixel [11]. A Radon transformed image represents the acquired data or *sinogram*. RT provides the analytical forward model in emission and transmission tomography, seismic imaging, and other disciplines. RT of a moving object, such as the heart, creates a blurred sinogram. The inverse Radon transformation (IRT) of a blurred sinogram introduces motion artifacts in the reconstructed image, and poses challenges to diagnostic applications in medical imaging [112–116]. Significant research efforts have been invested in addressing this challenge [114–120].

Cardiac and respiratory motion are particularly challenging in the medical context. Dedicated *gating* hardware are being developed for the purpose of synchronized data acquisition over individual motion-phases. Gating provides time-binned data from the same phase of a heart-beat or respiratory cycle. Typically, an electrocardiogram (EKG) is used for cardiac gating where data is time-binned over a few (6-10) cardiac phases [121–123]. For respiratory gating [124–126], many devices (e.g., infrared cameras) have been developed for similar data binning over the respiratory phases.

Subsequent image reconstructions over gated acquisitions provide motion-corrected images for each phase (cardiac or respiratory). Our work is motivated by a similar gated reconstruction of the probed object (e.g., heart), *without any need for hardware gating*. Our main contribution in this work is to propose a deep learning architecture based on Convolutional Neural Network (CNN), which is a type of artificial neural network (ANN) suitable for computer vision. We propose a specialized CNN model that uses so-called *attention* mechanism for reconstructing a single phase cardiac image from the respective motion-blurred sinogram. A deep learning model trained with motion blurred Radon transformed (RT) images will remove the need for any additional hardware or post-processing for the purpose, like using EKG for gating or some deep learning models for denoising (U-net).

In this chapter, we demonstrate how to address Inverse Radon Transform (IRT) with Artificial Neural Network (ANN) or Deep Learning, while performing motion correction. The purported application domain is cardiac image reconstruction in emission or transmission tomography where IRT is relevant. Our main contribution is in proposing an ANN architecture that is particularly suitable for this purpose. We validate our approach with two types of datasets. First, we use an abstract object that looks like a heart to simulate motion-blurred Radon Transform (RT). With the known ground truth in hand, we train our proposed ANN architecture and validate its effectiveness in Motion Correction (MC). Second, we used human cardiac gated datasets for training and validation of our approach. Gating mechanism time-bins data using electro-cardiogram (ECG) signals for cardiac motion correction. We have shown that the trained ANNs can perform better motion-corrected image reconstruction than most commonly used reconstruction techniques. We have compared our model against two existing ANN-based approaches to prove the former's superiority. This work provides an approach for reconstructing motion-corrected image that eliminates the need for

any hardware gating in medical imaging.

5.2 Methodology

Firstly, in subsection A, we describe how we have simulated motion affected images for training and testing. In the same subsection, we describe the human data used in our experiments. Subsection B discusses the structure of our proposed neural network, convolutional encoder-decoder extended with attention, or CEDA. In C, we elaborate on the statistical measures we have used for comparing the results from proposed CEDA to the other two approaches against ground truth or base-line images.

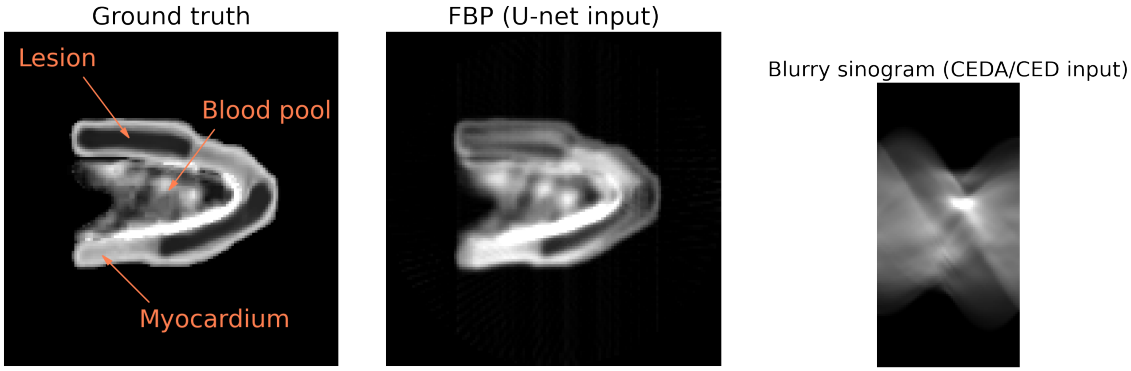
5.2.1 Data

5.2.1.1 Simulation

We start with the original data of a heart-shaped object that we refer to as *pseudo-heart*. There are three regions of interest (ROIs) on the apparent U-shaped object representing (1) infarcted tissue (lesion), (2) blood pool appearing as a cavity within the object, and (3) healthy myocardium (Fig. 5.1a). We have used random intensity variations on the pseudo-heart pixels to create a total of eighty original images. Fifty of these images are set aside for testing purposes and the other thirty images are further augmented with Affine transformations resulting in a total 60000 images for training the ANN. Scaling, shearing, translation, and rotational transformation models are applied to the images.

Each of the augmented training images is subjected to a set of arbitrary Affine motion to create a motion-blurred image. For blurring, we used multiple small Affine transformations which were averaged to create each motion-blurred image. This blurred image is then subjected to RT to produce the motion-affected sinogram, which acts

as the input layer to the ANN (CED and CEDA), while the corresponding ground truth original un-blurred image is used as the output layer. Fifty set-aside test images were not augmented but only blurred by Affine motions in a similar fashion as the test image set. Fig. 5.2 shows the work-flow of the data generation for the original pseudo-heart image. To simulate real data acquisition, Poisson noise was included to mimic the photon counting process.



(a) (b) (c)

Figure 5.1: An example pseudo heart: (a) ground truth, (b) FBP (c) blurry sinogram. (a) is the target output. (b) is the input of U-net. (c) is the input for CED/CEDA. Although not shown in the figure for clarity of visualization, Poisson noise has been added to the sinograms before they are used for training and testing the ANN model.

5.2.1.2 Human data

We have used retrospective data acquired from patients who underwent nuclear cardio stress-test with a Technetium-99m (^{99m}Tc) *sestamibi* tracer for cardiac SPECT studies. Twenty 3D datasets were provided by the Health First corporation's Holmes Regional Medical Center in Melbourne, Florida. Appropriate data-release consents were acquired at the clinic prior to imaging. The acquisition protocol for SPECT followed the established procedures and guidelines for cardiac studies. All data are completely anonymized at the clinic before our access. For each patient, we have (1) the sinogram, (2) gated reconstructed images of eight phases of the heart, and (3) un-gated recon-

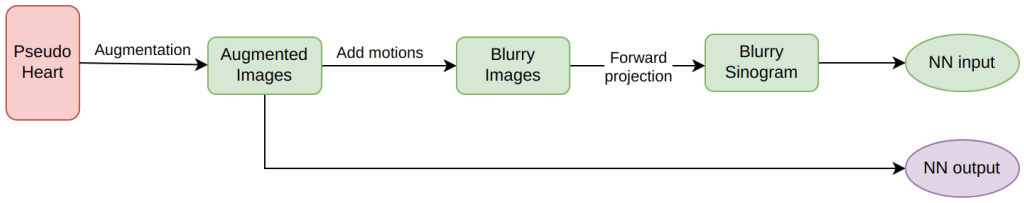


Figure 5.2: Work-flow for generation of training data with the pseudo-heart. After augmentation, a clear augmented image is used as the target output. We apply motion blur to these augmented images before forward projecting them to generate sinograms. Test data is generated similarly but without the augmentation step.

structed image that have blurring from the real cardiac motion. The MLEM algorithm was used for reconstruction.

The human data underwent necessary preprocessing steps before being used in our data analyses experiments, as described in this paragraph. Note that our goal is to reconstruct the motion-free cardiac image from the motion-blurred sinogram. For this purpose, we have two types of data: un-gated (motion affected) sinogram as the input, and gated (motion-free) MLEM reconstruction as the target output. In order to train a neural network, the sample size must be large enough. This required the training data to be augmented. For the same purpose (of obtaining a large data set), we used sliced 2D images. In total, 74160 augmented 2D images were created for training. Since we focus on the cardiac motion, we selected 2D-slices only around the heart. No augmentation was performed on the test set. The following describes how we generated the training and test data.

First, the un-gated sinogram was created by summing up all eight gated sinograms, and then Filtered Backprojection (FBP) was applied to obtain the reconstructed motion-blurred image. Next, MLEM reconstruction was performed on the gated sinogram for the *end-systole phase*, which had the smallest heart shape and maximum cardiac motion. Finally, we have chosen 206 2D-slices from FBP and MLEM reconstructions around the heart. These 206 slices were split into two groups: the first

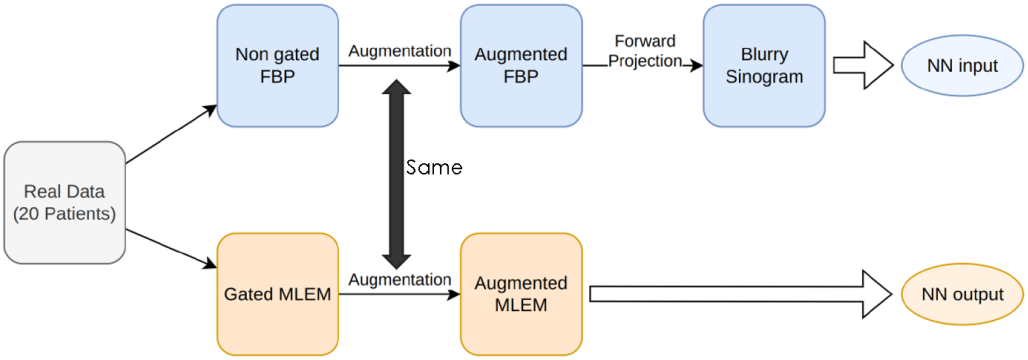
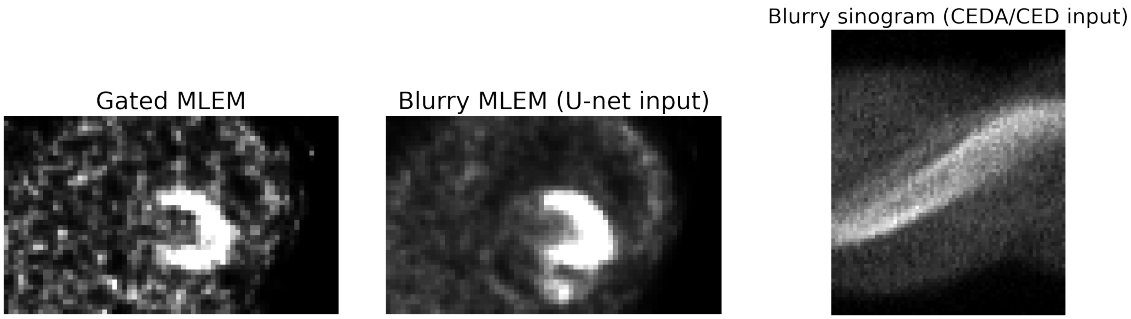


Figure 5.3: The process that displays how to create training data given the real human data.

103 slices were used for the training set creation, and the other 103 slices were set aside as the test set. Fig. 5.3 shows the process generating the training dataset. The un-gated FBP reconstructions were augmented by scaling, shearing, rotation, translation, and flipping. Each of the augmented images were forward projected to produce the un-gated motion affected sinogram. These un-gated sinograms were used as the training input. Similarly, the gated MLEM reconstruction went through the respective augmentation steps, so that each augmented motion-blurred sinogram (input) and the corresponding gated reconstructed image (output) correctly corresponded to each other in the augmentation step. No forward projection was needed for the MLEM since it was used as the target high-quality output for the ANN architectures. The 2D-slices in the test dataset underwent similar procedures to generate input and output pairs for testing. The only difference was that the 103 test images did not go through the data augmentation steps. Fig. 5.4 shows an example image of the human data for training.

5.2.2 The Neural Networks

Our proposed model (Fig. 5.5) is based on the Convolutional Encoder-Decoder (CED) [94] enhanced with the self-attention mechanism, which we call *Convolutional Encoder-Decoder with Attention* or CEDA [67, 68]. Originally, the CED was proposed by



(a) (b) (c)

Figure 5.4: An example human data: (a) gated MLEM reconstructed image, (b) blurry motion-induced MLEM, (c) blurry sinogram. Here (a) is the target output, (b) is the input for experiments with the U-net, and (c) is the input for the CED and the CEDA. The blurry MLEM (b) is obtained by performing MLEM reconstruction on the blurry sinogram, using the real system-matrix from the data acquisition protocol as in acquiring (c). For better visibility of the heart, all images presented here have been cropped around heart and adjusted for brightness, whereas the dimension of the actual images used in experiments were of 64×64 pixels.

Häggström, et al [94]. The conventional CED architecture contains two parts: the encoder and the decoder. The encoder is similar to a CNN, like LeNet-5 [32] [35] and VGG-16 [127]. Our encoder takes the sinogram as the input and then compresses it into a one-dimensional vector. The second component, the decoder, is the reverse of the encoder and decompresses the encoding to reconstruct the image (Fig. 5.5).

In CEDA encoder, there are six convolutional components. Each component has two convolution-batch normalization units: a convolutional layer followed by a batch normalization layer. The structure of the decoder is similar to the encoder, but the decoder uses deconvolutional layers instead of convolutional layers. The self-attention component is added after the 4th and 5th convolutional components in both the encoder and decoder. We used *Adam* optimizer with the learning rate $r = 0.002$. Dropout layers are added after the encoder and decoder with the rate of 0.2. Additionally, to prevent overfitting, we applied an inverse time decay for the learning rate schedule. After 10 iterations, the learning rate is halved. In total, there are $3,192,869$ number of parameters in CED, and $10,201,316$ in CEDA.

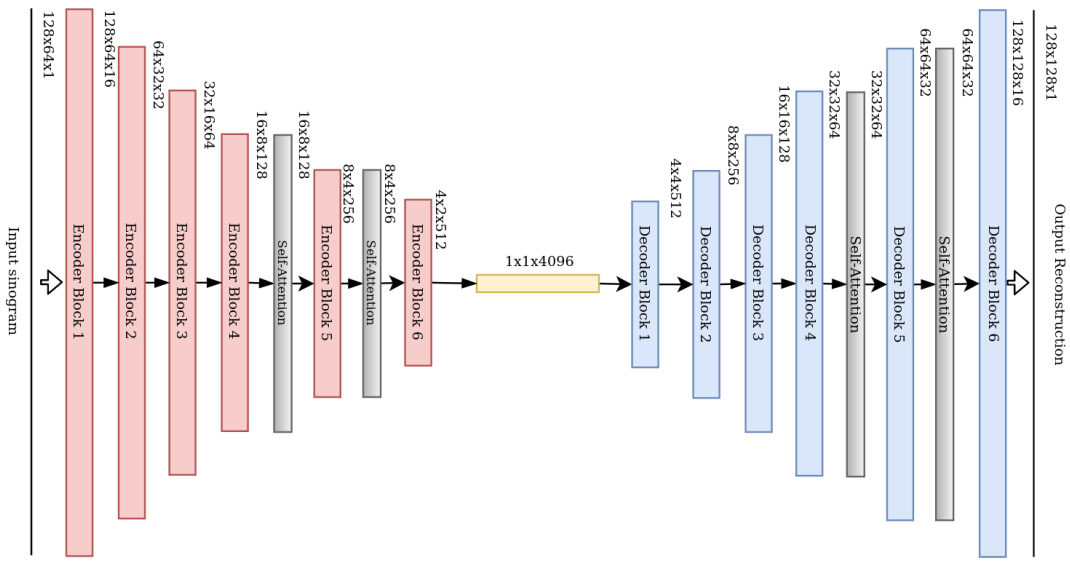


Figure 5.5: The architecture of the proposed CEDA. Each encoder block contains two convolutional layers. Each decoder block contains one convolutional transpose layer followed by two convolutional layers. Every convolutional layer in both the encoder and the decoder is followed by a batch normalization layer and a Leaky ReLU layer. Two self-attention layers are used in the encoder and the decoder.

In addition to comparing our model against the conventional CED, we have also compared CEDA against a U-net for deblurring. The architecture of U-net contains a contracting path and an expansive path. Further, there are several "bridges" to connect the features in the contracting path and the upsampled outputs in the expansive path. The U-net used the blurry MLEM reconstruction as the input and the gated MLEM as the output. The U-net transforms a motion blurred image to a deblurred version of the image. It cannot perform image reconstruction directly from a sinogram to an image which involves IRT.

5.2.3 Measures used for comparison

We measured the performance of three models (CEDA proposed by us, the original CED, and a deblurring U-net [128]) against known ground truth in simulation and the gated (motion-free) MLEM reconstruction as the base-line for human data. The

performance of CEDA is compared against the other two.

We utilize *visual information fidelity* (VIF) [129] as a measure to compare the quality of reconstructed image. The VIF is shown to be very close to the human assessment of image quality [130]. It uses a channelized information-theoretic concept of mutual information to quantify the information shared between the reference image (ground truth) and the referred image (prediction) where *Channels* may be the *wavelet* bands. It assumes that the shared information is highly related to the fidelity of the visual quality. In simple terms, VIF is the ratio of $I(C;F)$ and $I(C;E)$, where $I(x;y)$ is the mutual information between image x and image y. C is the reference image, E is the output of the human visual system (HVS) given the reference image C, and F is the output of HVS given the referred (presumably, corrupted) C. In general, VIF lies between 0 and 1, and the higher the VIF, the better it is for a human observer. If two images are exactly the same and no information is lost, the VIF would be 1.

We have also used the signal-to-noise ratio (SNR, eq. (5.2)) and contrast-to-noise ratio (CNR, eq. (5.3)) to compare between reconstructed images from different models. In both equations, μ_{ROI} is the average pixel value of the ROI, μ_{BKGD} is the average pixel value of the background (the region outside the ROI),and σ_{BKGD} is the stdv of the background. A strong SNR enables the visual sensitivity to easily distinguish the desired image signal over the background noise. The CNR subtracts the μ_{BKGD} before taking the ratio, and is beneficial in scenarios such as haze or motion-blur. Ultimately, CNR has similarity to those of human visual evaluations.

$$VIF = \frac{I(C;F)}{I(C;E)} \tag{5.1}$$

$$SNR = \frac{\mu_{ROI}}{\sigma_{BKGD}} \tag{5.2}$$

$$CNR = \frac{\mu_{ROI} - \mu_{BKGD}}{\sigma_{BKGD}} \tag{5.3}$$

5.3 Results

We ran our experiments on Intel(R) CPU Core(TM) i7-9700K with clock-cycle @ 3.60GHz, and memory 32GB RAM. We used an NVIDIA GPU Titan XP with 12GB memory. The ANNs were developed with the Keras [131] library with Tensorflow [132] back-end. Keras is an open source library which provides high-level Python APIs for building the ANN. In the following we describe our results from each experiment.

5.3.1 Simulation

Fig. 5.6 shows some examples of output reconstruction of the human data from different models. The first row is from CED reconstruction, the second row is CEDA reconstruction, the third row is from the U-net deblurring, and the last row is from conventional reconstruction algorithm considered as base-lines. The output of U-net deblurring is not as clear as those from CED and CEDA. The U-net significantly failed to remove motion-blur in most of the images, and some of the regions are actually further dilated (the third row in Fig. 5.6).

Fig. 5.7 shows the comparison measures over the test dataset. They are computed over a fixed region around the heart, since the background is unimportant and confuses the measures for cardiac studies. The ordering of the data elements on the X-axes is

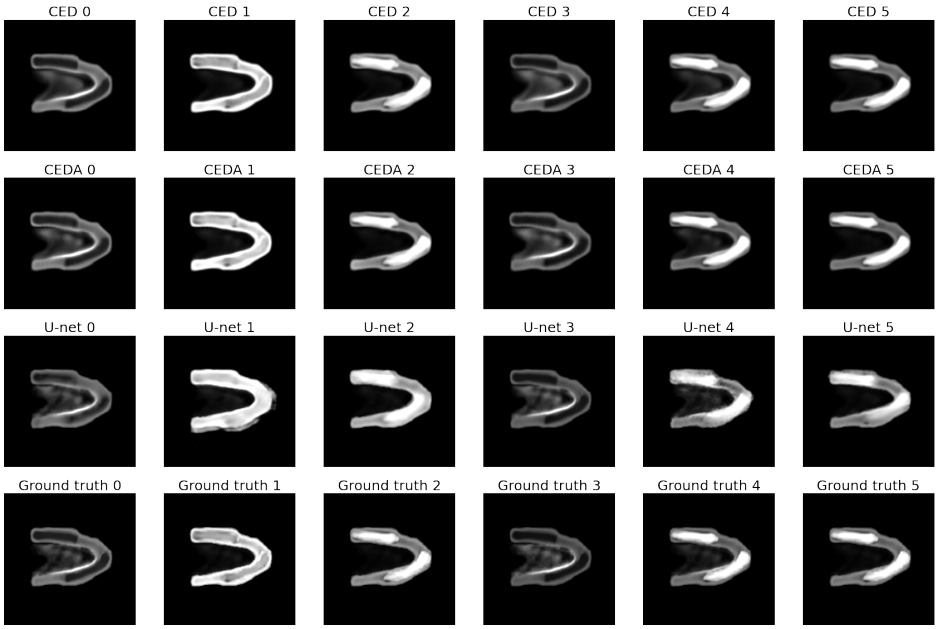


Figure 5.6: Examples of reconstructed images for pseudo-heart data from different models. First row is from CED reconstruction, second is from CEDA reconstruction, and the third row is from U-net deblurring. The last row shows the corresponding ground truth (clean motion-free) images.

arbitrary. The VIF is measured against the ground truth, and the SNR and CNR are measured for each model-generated image. According to the VIF (Fig. 5.7a), the CED and CEDA have better quality reconstructions for human observers than those from the U-net. Note that U-net performs deblurring over conventionally reconstructed blurred images. The mean VIF of CED is 0.61 ± 0.059, CEDA 0.62 ± 0.049, U-net 0.47 ± 0.049 (Fig. 5.8a). A Kruskall-Wallis test performed for each of these tests comparing CEDA to U-net and CEDA to CED. Results were significant for the U-net and CEDA comparison for all of the similarity measures. For VIF with CEDA (*Median or Mdn = 0.6097*) and U-net (*Mdn = 0.4522*), *H(2)* = 58.6608, $p < .00001$.

The ROI used to compute SNR was the region of myocardium containing the lesion. For SNR (Fig. 5.7b), CEDA has the best performance. The mean SNR of CED is 30.44 ± 10.665, whereas that of CEDA is comparatively higher, 32.12 ± 11.900. The SNR values from the U-net are much lower, 20.88 ± 9.717 (Fig. 5.8b). A Kruskall-

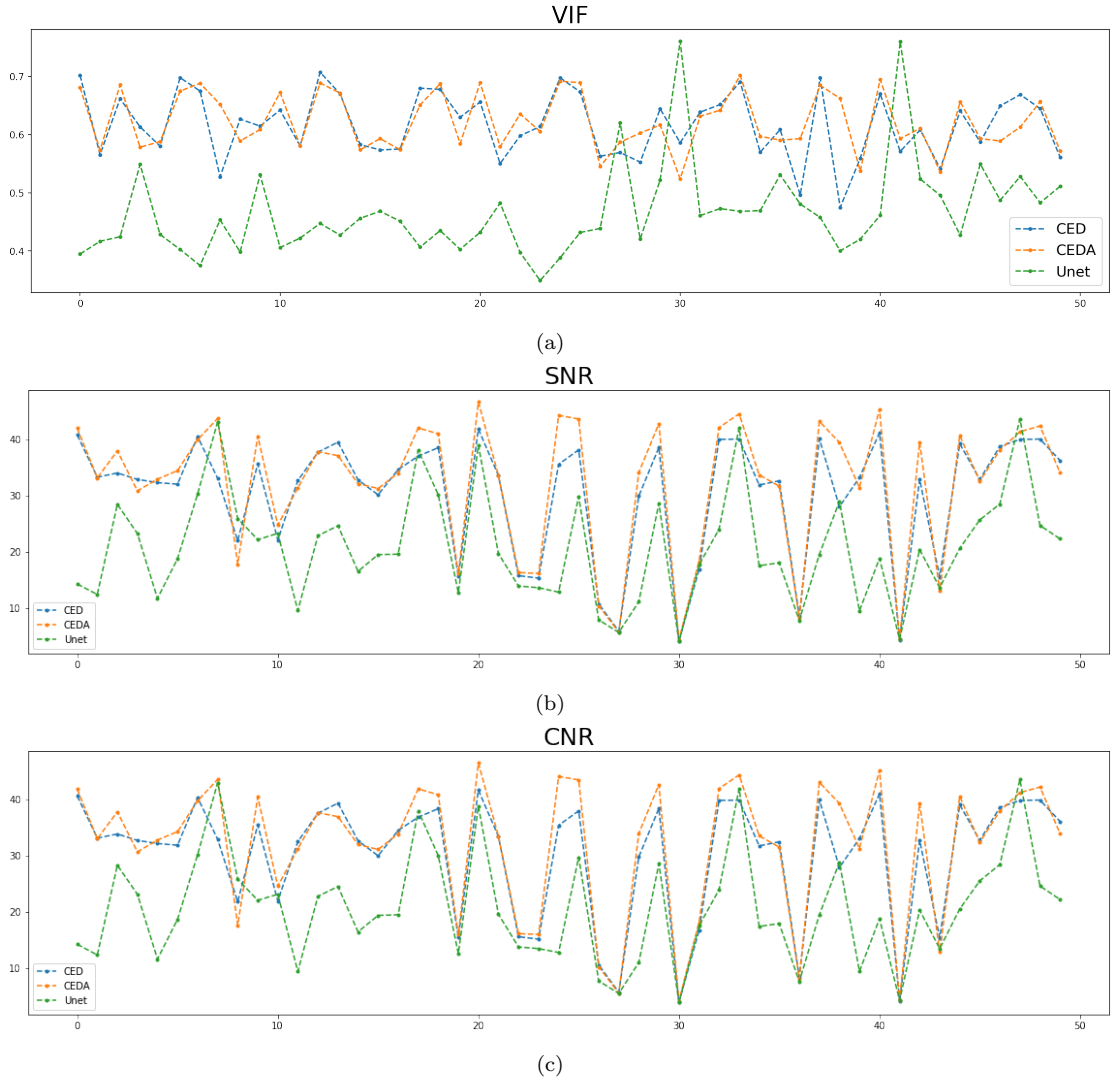
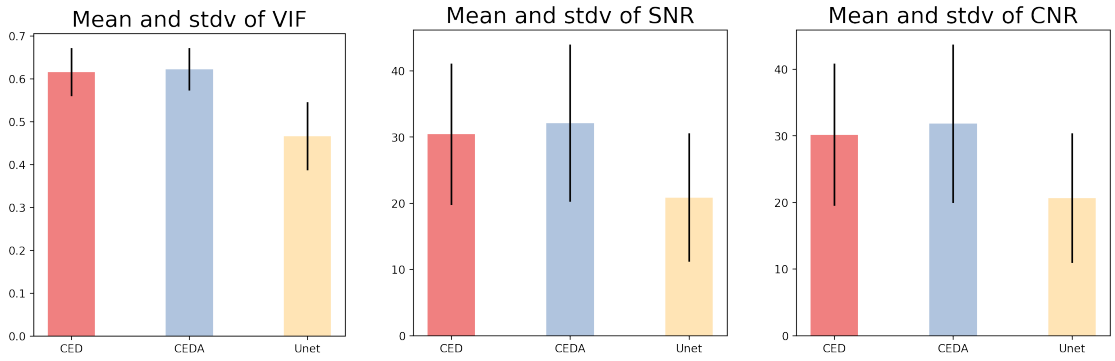


Figure 5.7: Plots of the three different measurements used to compare the performance of different models against the ground truth in simulation. In each plot, X-axis represents the arbitrarily ordered indices of test images, and the Y-axis represents the values of the corresponding measurements.

Wallis test of SNR for CEDA and U-net $H(2) = 21.5894$, $p < .00001$, showed again that CEDA ($Mdn = 34.16$) out-preforms U-net ($Mdn = 19.68$).

Our third comparison measure was CNR, which utilizes the same ROI as used for SNR. According to the result of CNR (Fig. 5.7c), CEDA also appears to have performed the best. The mean CNR of CED is 30.16 ± 10.667, CEDA 31.84 ± 11.902, and U-net 20.65 ± 9.719 (Fig. 5.8c). Similar to SNR, the Kruskall-Wallis test for CNR



(a) (b) (c)

Figure 5.8: Pseudo-heart comparison of average measures over all test data: (a) VIF, (b) SNR, and (c) CNR values as computed from the outputs of different models (CED, CEDA, and U-net) against the ground truth.

$H(2) = 21.4614$, $p < .00001$, showed a significantly different and better performance for the CEDA ($Mdn = 33.89$) as compared to U-net ($Mdn = 19.47$).

5.3.2 Human data

Fig. 5.9 shows some example output from CED (first row), CEDA (second row), and U-net (third row). We also show the gated MLEM reconstruction (last row) that acts as a base-line for comparison, in the absence of any ground truth as in the case of experiments with simulation data. The U-net reconstruction appears similar to the gated MLEM reconstruction ascertaining that U-net can perform motion correction in real data. However, CED and CEDA reconstructions appear smoother and clearer. On careful observation, the CEDA appears to have enhanced the myocardium segment, giving it a brighter and cleaner appearance, justifying the attention mechanism introduced in the network.

To confirm our visual analyses of output from the three models, we have measured VIF of each reconstructed image against the base-line conventionally reconstructed image, and SNR and CNR of each model-generated image. Fig. 5.10 and the Fig. 5.11 show the measures over the test images. The SNR and CNR are measured over a small

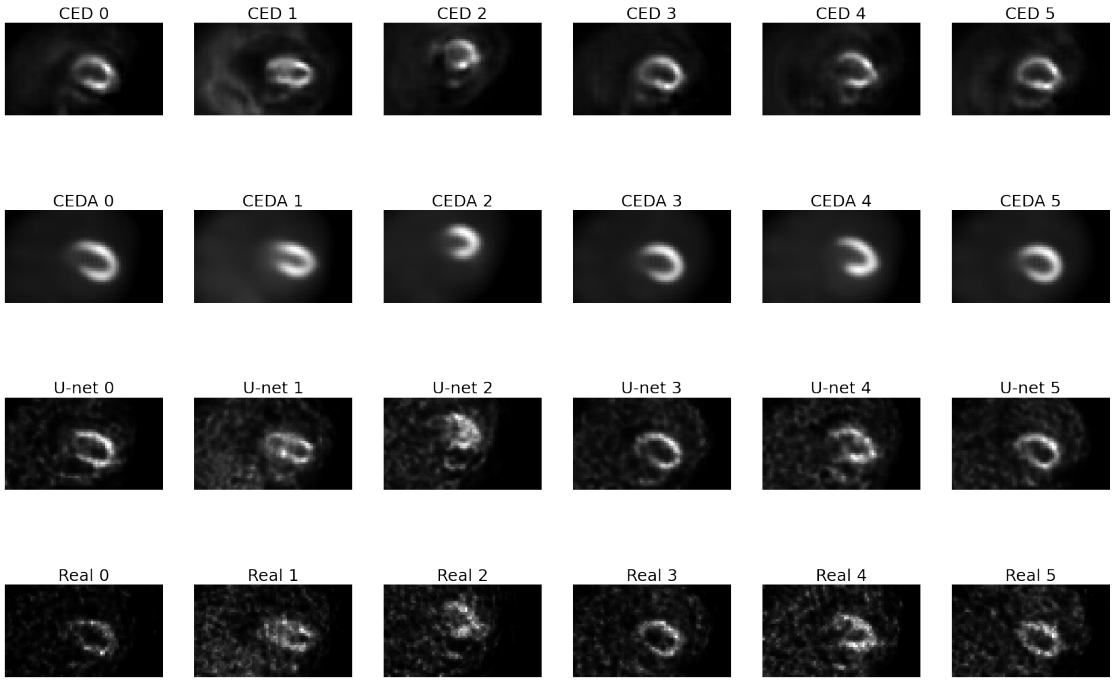


Figure 5.9: Random sample output of human data from different models. The first row is from CED, the second row is from CEDA, the third row is from U-net, and the last row shows the base-line iterative reconstructions from the gated MLEM.

box around the heart or the ROI, and the background is disregarded.

Fig. 5.10a shows that U-net attained the best VIF in overall, while CEDA remains better than CED. The mean VIF of CED is 0.13 ± 0.033, CEDA 0.15 ± 0.040, U-net 0.20 ± 0.053 (Fig. 5.11a). For every test image (out of 103 test images in total), all of U-net's deblurred images provided higher VIF than CEDA reconstruction. When comparing CEDA and CED, 95 images were reconstructed by CEDA with higher VIF scores than that from CED. A Kruskal-Wallis test was performed to measure the relative performance between CEDA and CED as well as CEDA and U-net. Results were significant for CEDA comparison to CED and CEDA comparison to U-net. For VIF, $H(2) = 13.2214$, $p = .00028$, for CEDA ($Mdn = 0.14564$) and CED ($Mdn = 0.13010$). Additionally, for VIF, $H(2) = 52.6987$, $p < .00001$, for CEDA and U-net ($Mdn = 0.19347$), showing U-net's effectiveness over CEDA.

For the SNR (Fig. 5.10b, CEDA provides the best performance. The mean SNR

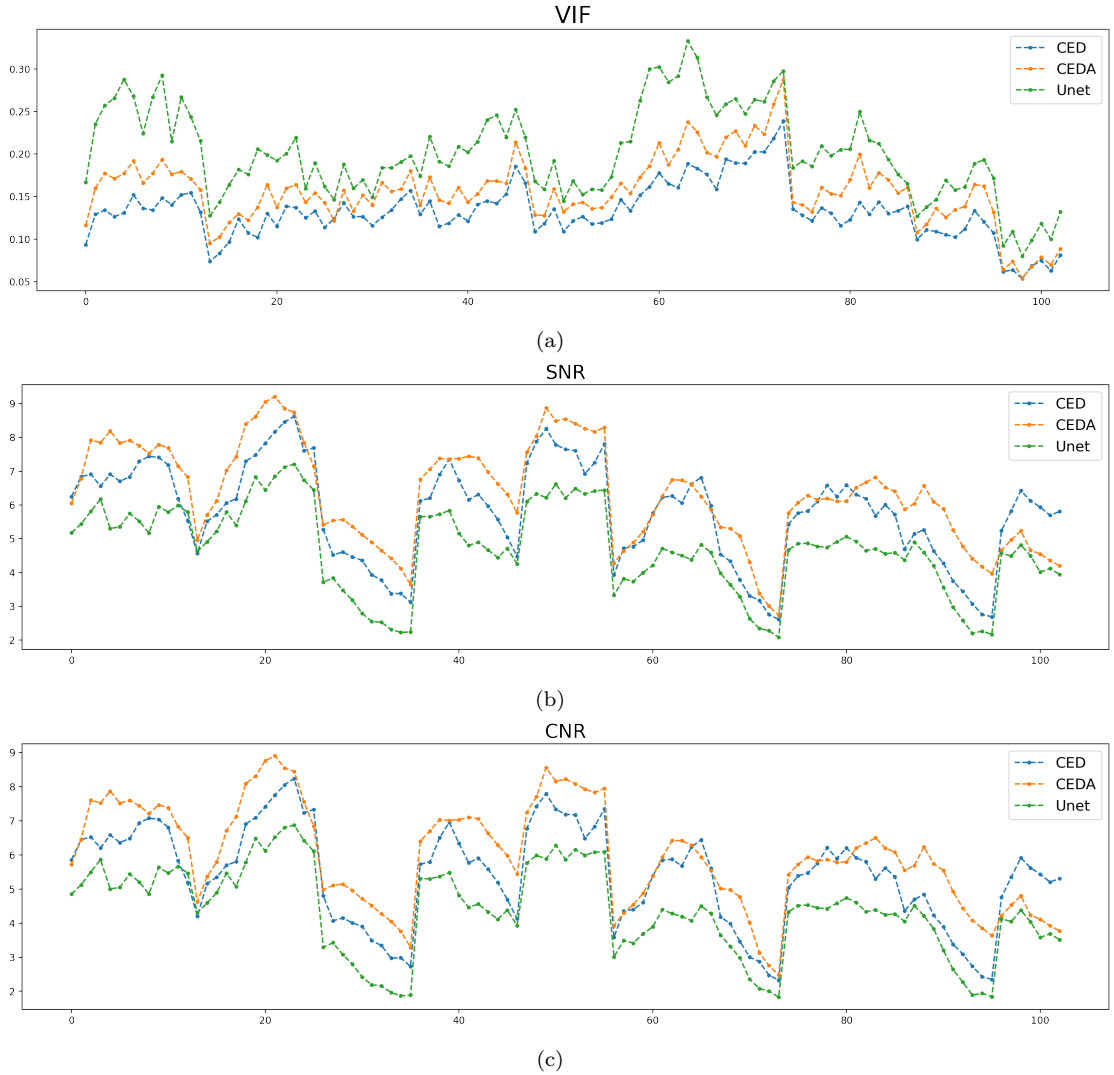
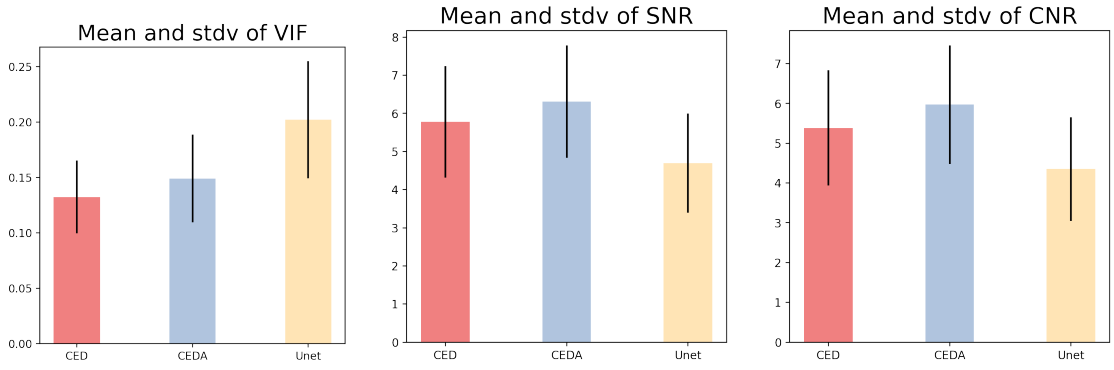


Figure 5.10: Comparison of different outputs using VIF, SNR, and CNR for human data. The reference image is the gated MLEM (end-systole) reconstruction. Note that U-net deblurs the motion-corrupted MLEM from conventional reconstruction as its input. Both CEDA and CED directly used motion-corrupted sinogram as their input. In each plot, X-axis represents the arbitrarily ordered indices of test images, and the Y-axis represents the values of the corresponding measurements.

of CED is 5.78 ± 1.459, CEDA 6.31 ± 1.478, U-net 4.69 ± 1.299 (Fig. 5.11b). When comparing the SNR between CEDA and U-net, all of CEDA's reconstructions are better than those of U-net. Between CEDA and CED, there are 85 images where CEDA obtained a higher SNR than CED. The CNR measurement (Fig. 5.10c) also shows CEDA performing the best. The mean CNR of CED is 5.39 ± 1.446, CEDA



(a) (b) (c)

Figure 5.11: Overall mean and stdv from different models over human data, for (a) VIF, (b) SNR, and (c) CNR. Vertical lines on the bars indicate the stdv of each measure.

5.97 ± 1.492, U-net 4.35 ± 1.302 (Fig. 5.11c). As with SNR, all of CEDA's output images are better than those of U-net, and there are 87 images where CEDA has a higher CNR than CED. Kruskall-Wallis test for SNR, $H(2) = 49.1953$, $p < .00001$, showed that CEDA ($Mdn = 6.2633$) performed better than U-net ($Mdn = 4.7145$). Similarly, the Kruskall-Wallis test for SNR $H(2) = 5.1255$, $p = .02358$, showed that CEDA also performed better than CED($Mdn = 6.0039$). The Kruskall-Wallis test for CNR, $H(2) = 48.184$, $p < .00001$, showed that CEDA ($Mdn = 5.9362$) significantly performed better than U-net ($Mdn = 4.3839$), while, the Kruskall-Wallis test for SNR, $H(2) = 6.4209$, $p = .01128$, showed that CEDA also performed better than CED($Mdn = 5.61823$).

5.4 Discussion

We discuss below our results from the simulation and the human data separately for three statistical measures used for comparison. To compare these measures from different models, we use a non-reflexive *relative difference* (RD) measure, as defined below.

$$RD_S(M_a, M_b) = \left| \frac{M_a - M_b}{\frac{1}{2}(M_a + M_b)} \right| \tag{5.4}$$

where S is the respective statistical measure like VIF, CNR, or SNR), and M_a and M_b are the measures from the two compared network models like U-net, CED, or CEDA).

5.4.1 Simulation data

5.4.1.1 VIF

Since VIF is the ratio of the information content in the test image and the reference ground truth image, a higher VIF indicates less information loss. Relatively lower VIF values for U-net indicated that it performed worse than CED and CEDA. Hence, in most of the cases U-net has produced images with less information content than the reconstructions by CED and CEDA. Some of the deblurred images from U-net are visibly noisier (Fig. 5.6) than the original input blurred reconstructions. U-net appears to be less adept at removing the Affine motion blur as introduced in the simulation. A possible explanation is that the connecting bridges in U-net between downsampling and upsampling paths transmitted the blur information between the input and the output. Also to be noted, the VIF scores from both CEDA and CED are more stable than those from the U-net as seen by a lower standard deviation.

Comparing specifically (Fig. 5.7a) U-net against CED/CEDA, 47 images reconstructed by CED and CEDA have higher VIF scores than those from the U-net, demonstrating that the U-net performs worse than CED and CEDA. As we compare VIF scores between CED and CEDA, CEDA provides 27 images with better VIF than CED. Furthermore, based on mean VIFs (Fig. 5.8a), CEDA illustrates a slightly better performance than CED.

The $RD_{VIF}(CED, CEDA)$ reveals that CEDA is 5.22% better than CED for the

27 images where CEDA has higher VIF values, while it is 3.81% worse for the rest of the images where CEDA has lower VIF. Table 5.1 displays the $RD_{VIF}(CED, CEDA)$ of all the models.

Table 5.1: VIF RD of Pseudo-heart between different models

Models		# of images	RD
U-net vs. CED	U-net better	3	20.94%
	CED better	47	31.38%
U-net vs. CEDA	U-net better	3	22.32%
	CEDA better	47	32.59%
CEDA vs. CED	CEDA better	27	5.22%
	CED better	23	3.81%

5.4.1.2 SNR and CNR

The figures Fig. 5.7b and Fig. 5.7c display the mean and stdv of SNR and CNR, respectively. Both CED and CEDA perform better than U-net by these measures, while CEDA performs the best of the three. Furthermore, inspecting SNR and CNR of individual reconstructed images reveals that CED performs better for 41 images than U-net, and CEDA performs better over 47 images than the U-net. For 33 images, CEDA has higher SNR and CNR than CED. Table 5.2 and Table 5.3 summarize the RD comparisons of all the different models for SNR and CNR.

Table 5.2: SNR RD of Pseudo-heart between different models

Models		# of images	RD
U-net vs. CED	U-net better	9	8.25%
	CED better	41	47.14%
U-net vs. CEDA	U-net better	3	15.72%
	CEDA better	47	44.77%
CEDA vs. CED	CEDA better	33	9.07%
	CED better	17	5.18%

Table 5.3: CNR RD of Pseudo-heart between different models

Models		# of images	RD
U-net vs. CED	U-net better	9	8.25%
	CED better	41	47.14%
U-net vs. CEDA	U-net better	3	15.72%
	CEDA better	47	44.77%
CEDA vs. CED	CEDA better	33	9.17%
	CED better	17	5.23%

5.4.2 Human data

5.4.2.1 VIF

In experiments with human data, U-net deblurring achieves the highest VIF in all the 103 testing images compared to either CED or CEDA. On average, the VIF of U-net is 41.27% higher than CED and 30.52% higher than CEDA. CEDA has better image quality than CED according to VIF (Table. 5.4). There are 95 images reconstructed by CEDA which are 12.54% higher than CED. The rest of the 8 images where CEDA has lower VIF score, the CEDA is 4.92% worse than CED.

Table 5.4: VIF RD of human data between different models

Models		# of images	RD
U-net vs. CED	U-net better	103	41.27%
	CED better	0	NA
U-net vs. CEDA	U-net better	103	30.52%
	CEDA better	0	NA
CEDA vs. CED	CEDA better	95	12.54%
	CED better	8	4.92%

5.4.2.2 SNR and CNR

Both CED and CEDA have better SNR and CNR than U-net. Comparing between U-net and CED, we find only 2 images where U-net has a higher SNR and CNR. For

the remaining 101 images, CED gets better SNR and CNR, with those of CEDA being even better, since it has the highest score for all the 103 images compared to those from U-net. For the comparison between CEDA and CED, CEDA leads to superior results, with 85 images with better SNR (RD 13.83%) and 87 images with better CNR (RD 15.40%). The results are summarized in (Table. 5.5 and Table. 5.6). We purport that CEDA's superior performance can be ascribed to its directed focus on the ROI and suppression of the background noise.

Table 5.5: SNR RD of human data between different models

Models		# of images	RD
U-net vs. CED	U-net better	2	2.90%
	CED better	101	21.97%
U-net vs. CEDA	U-net better	0	NA
	CEDA better	103	30.71%
CEDA vs. CED	CEDA better	85	13.83%
	CED better	18	11.15%

Table 5.6: CNR RD of human data between different models

Models		# of images	RD
U-net vs. CED	U-net better	2	3.75%
	CED better	101	22.80%
U-net vs. CEDA	U-net better	0	NA
	CEDA better	103	33.00%
CEDA vs. CED	CEDA better	87	15.40%
	CED better	16	12.62%

5.4.3 Analysis

From the RD analyses we observed that by all measures and data types (simulation and human), direct reconstructions of motion-affected sinograms by CED and CEDA are uniformly better than deblurring with U-net post-processing, except by the VIF measure on human data where U-net performed significantly better. Overall, our proposed

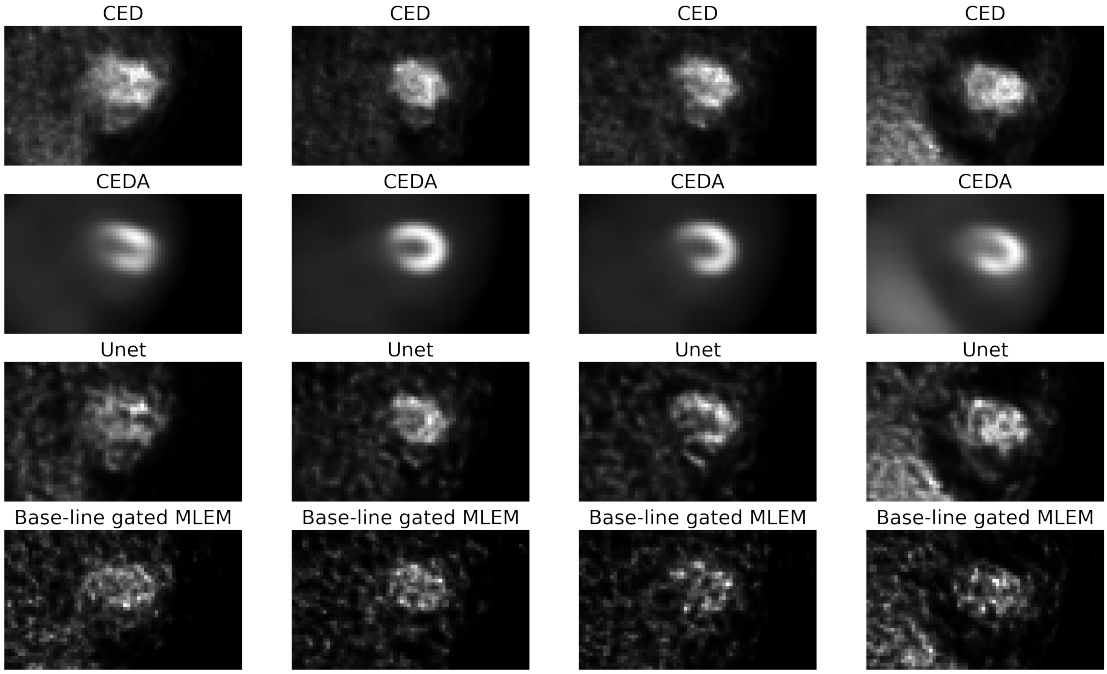


Figure 5.12: There are some gated MLEM reconstructions (last row) which are not very clear due to the low counts. As we can see, U-net tries to be similar to the gated MLEM, while CED/CEDA tries to improve it by inferring the shape.

model CEDA performed the best of the three models, aside from the VIF measure for clinical data with U-net. A better VIF indicates that U-net has a better capability to learn channel distortions caused by cardiac motion, whereas direct reconstruction models CED and CEDA are better adapted for noise removal in ROI (as indicated with the higher SNR and CNR). However, CED and CEDA produced clearer reconstructed images (Fig. 5.9) as compared to base-line MLEM reconstructed images, while U-net maintained visual similarity, as measured by VIF, with target MLEM images. Furthermore, since the gated projection data has relatively poor counts, the gated reconstruction may not necessarily show a clear shape (Fig. 5.12). For those reconstructions, U-net tries to imitate the target reconstructed image, while CEDA tries to reconstruct a smoother noise-free and motion-deblurred image with the correct shape. One should also note that, as a post-processing method, U-net needs the image reconstruction from a conventional algorithm first, and only then can it perform deblurring. Our proposed

model CEDA, and the comparable model, CED, can directly use the noisy motion blurred sinogram as the input and reconstruct a motion-free reconstructed image. As a result, it is easier and faster to use than U-net. For example, iterative reconstruction took 0.7 sec, followed by deblurring (inferencing by the trained U-net model) of 0.02 sec, whereas CEDA inferencing (direct reconstruction) from sinogram took 0.04 sec, about 18 times faster.

5.5 Conclusion

The objective of this work was motion correction while performing IRT from the motion blurred Radon transformed image. We developed a new deep learning architecture called CEDA for this purpose. The model was tested with two types of datasets, a simulated and a human cardiac dataset. Deep learning-based IRT has many implications, one of which is faster image reconstruction. For example, it learns the data acquisition (or underlying physics) model and the motion-model from a training set, and does not need any additional hardware (e.g., gating) or software (e.g., system-matrix generation). We validated our proposed model against ground-truth in simulation, and the traditional MLEM reconstructed base-line image in clinical data. We compared the quality of our approach against two other deep learning-based approaches.

A future direction of this work involves training and validating the model directly with 3D images rather than with 2D slices as has been done in this work. We could not use 3D presently for two primary reasons: (1) the required computational resources for training with 3D images are exorbitant, not only for a research set up, but also in a clinical scenario, and (2) the number of training and validation data points needed in a 3D work is far more than what is easily available from clinics. Currently, a standard practice in the literature is to slice the 3D images around the ROI (i.e the heart) in

sufficient numbers of 2D data sets, as we did to prove the concept. Looking forward, we aim to acquire more 3D data for better training of the model and for validating its motion correction capability in a diagnostic setting, e.g., in detecting cardiac infarction. Also, we intend to apply this technique in non-medical areas where motion affects IRT, e.g., in astrophysics.

Chapter 6

PET Imaging of Mouse

In this chapter, we answer the question regarding how to reconstruct the image when data is not sufficient for ANN training. The data we used is the pre-clinical mouse data, which is provided by Cardiff university. Our main task is to reconstruct the motion-free image from the noisy sinogram, without using any real data to train the ANN model. Before doing that, we tested an adapted CED model to reconstruct the clear image from the Radon transform of the corresponding noisy image.

6.1 Prior Work

The data we used is the PET scan of a healthy mouse. During this experiment, we mainly focus on the cardiac image. Since this data doesn't contain sinograms, we applied the Radon transform on the dynamic scan to obtain sinogram, and chose the gated scan to be the target output of the neural network. Because the dynamic scan is noisy, we want to see if the neural network model can produce motion-free reconstructions given the noisy images. To test the accuracy of our approach, we selected one of the six gates that exhibited the most deformation of the myocardium in

comparison to the reference gate at end systole. Some common image transformation methods were applied on the these images for augmentation. Then the sinograms of this augmented data were generated by Radon transform. The neural network model we trained is CED. It takes those motion corrupted (dynamic) sinograms as input, and uses the Gated scan as the target output. Our motivation is to train the CED to take the noisy sinograms as input, then produce the clear reconstructions void of motion degradation.

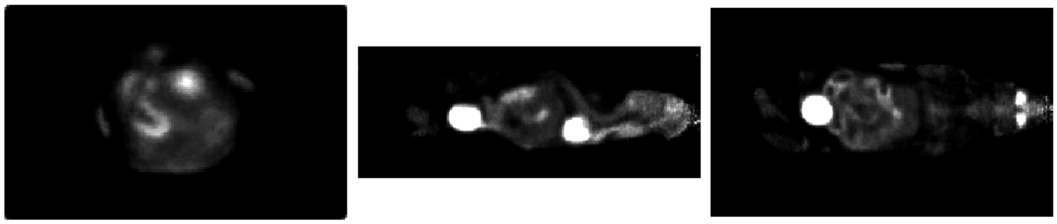


Figure 6.1: The dimension of the dynamic scan is 150x94x245. Fig. 6.1 shows a certain slice of the scan in three different anatomic planes. The left one is the axial plane, the middle one is the sagittal plane, and the right is the coronal plane. The corresponding dimension of these images are (from left to right): 95x150, 94x245, 150x245.

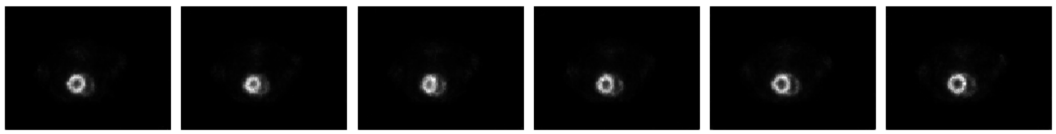


Figure 6.2: The dimension of each gate is 127x95x245. Here we picked up the 156-th slice from each gate. Therefore, the dimension of the slices is 95x127. Note that the third one (gate) is the most stressed one. We chose this gate as the neural network target output.

Fig 6.1 shows the dynamic scan of this mouse. The dimension of it is 150x94x245. The gated scan we have contains six gates, with the dimension of each 127x90x245. Fig 6.2 shows the same heart slice but coming from six different gates. Cardiac movement artifacts are one of the major image degrading factors that require reduction. Therefore, we take some slices around the heart of the three main anatomical planes (coronal, sagittal, and transverse). In total, we obtain 40 slices (coronal 12 slices, sagittal 11 slices, transverse 17 slices). Due to the computational consideration, all the

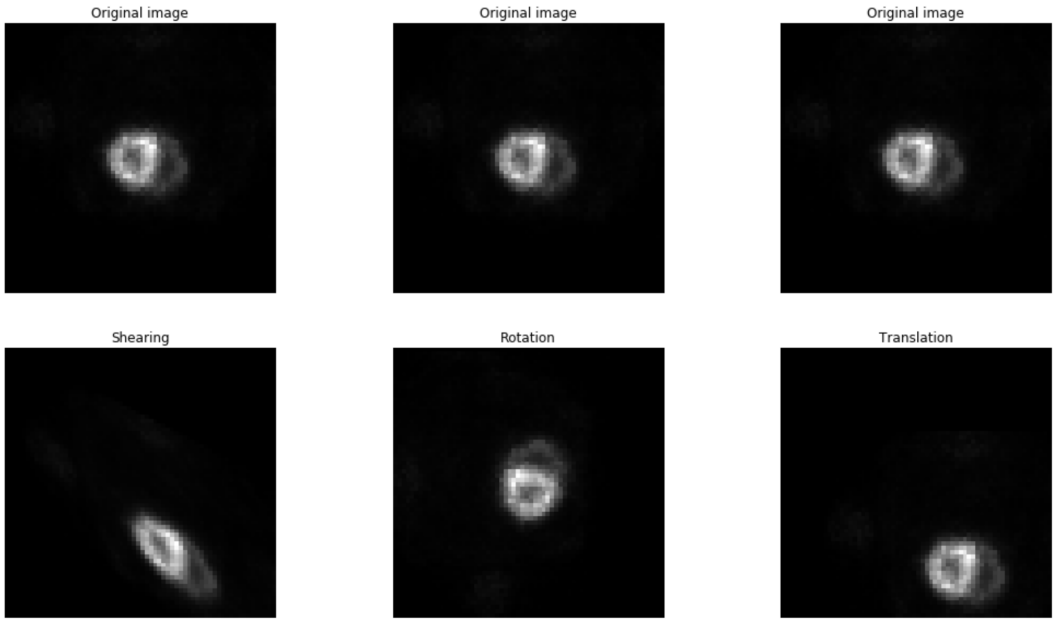


Figure 6.3: Examples of the image transformation. The images in the first row are the original images (same slice). The second row contains the images after transformation. From left to right, they are: shearing, rotation, and translation.

slices are cropped to 64x64 pixels. Firstly according to the slice indices of the three different planes, determine the center of the heart (center of the bounding box). Then decide the boundary of the cropped image according to this center (± 32). The 2D slices of the gated scan will be directly used as the target output. For the slices of the dynamic scan, we need to do one more Radon transform step to produce the sinograms as the input. However, before Radon transform, both gated and dynamic slices need augmentation.

Originally, there are 40 slices. We split them such that alternating slices from the same plane served as testing; i.e, first slice for testing, second one for training, third one for testing, fourth one for training, and so on. This results in two sets of slices: one set 19 slices and the other set 21 slices. We use the set with 19 slices as the training set and perform augmentation; the set with 21 slices serves as the testing set. We use shearing, rotation and translation to augment the training images. Equation (6.1, 6.2,

and 6.3) show these transformations in matrix form. $[x, y]$ is the original coordinate of a certain image pixel. $[s_x, s_y]$, $[r_x, r_y]$, and $[t_x, t_y]$ are the coordinates after shearing, rotation and translation. c_x and c_y are the shearing factors. θ is the rotation angle. $[v_x, v_y]$ is the translation vector. We also use horizontal and vertical flipping for the augmentation.

$$\begin{bmatrix} s_x \\ s_y \end{bmatrix} = \begin{bmatrix} 1 & c_x \\ c_y & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{6.1}$$

$$\begin{bmatrix} r_x \\ r_y \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{6.2}$$

$$\begin{bmatrix} t_x \\ t_y \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} \tag{6.3}$$

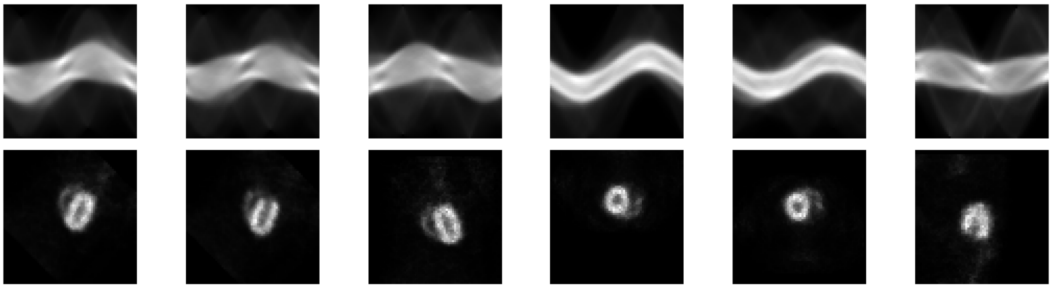


Figure 6.4: Some augmented images and the corresponding sinograms. The first row contains the sinograms and the second row displays the gated images.

Fig. 6.3 shows an example of images after shearing, rotation and translation. Shearing factors are $c_x = 0.2, c_y = 0.5$. Rotation angle is 90 degree. The translation vector is $[10, 20]$. Note that we need to apply the same transformation on both the dynamic images and gated images. There is one more step for the dynamic images: use the Radon transform to generate the sinograms of these augmented dynamic images. Fig. 6.4 show some of the images after augmentation. The first row shows the sinograms, and the second row shows the corresponding gated images. Finally, the training data

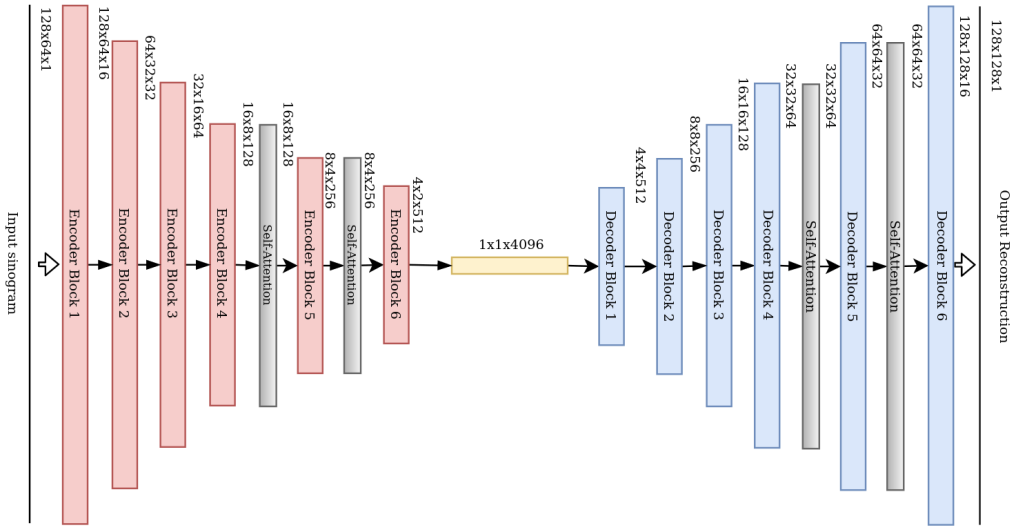


Figure 6.5: Convolutional Encoder-Decoder architecture. The numbers around the input and output images are the dimension. The numbers near the cubes are the dimension of the feature maps.

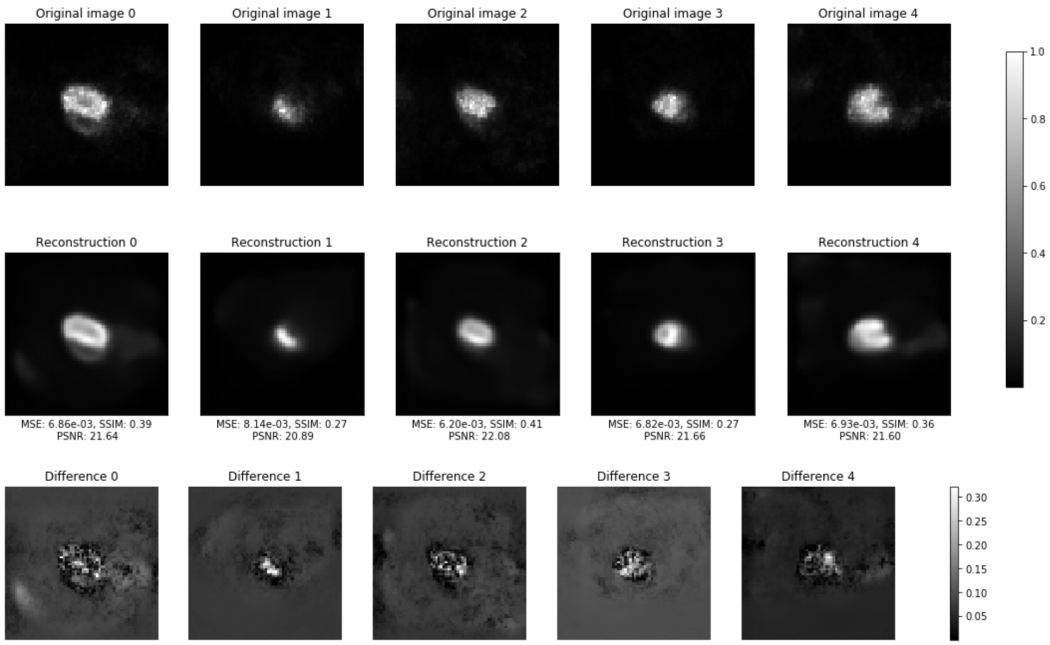


Figure 6.6: Some results. First row shows the original gated images (target output). Second row is the reconstructed output from CED. Third row is the difference image between original gated image and the reconstruction. Each image in the second row also shows some statistics (MSE, SSIM, and PSNR) compared to its corresponding gated image.

set has 201571 images as well as the sinograms.

The model we use is called CED, which is adapted from DeepPet [94]. The input is the sinogram with a size of $64 \times 64 \times 1$. The output reconstructed image has the same size ($64 \times 64 \times 1$). The model has two components: an encoder and a decoder. The encoder is just a common convolutional neural network (CNN). It has several convolutional units. Each unit has one convolutional layer, followed by one batch normalization layer and one leakyReLU activation layer. The filter size of the first 6 convolutional layers is 5×5, the filter size of rest of the units are 3×3. The output of the encoder has 64 feature maps of size 4×4. The decoder takes these encoding messages and performs upsampling to reconstruct the gated images. Therefore, the decoder appears as a mirror of the encoder. Each unit in the decoder is very similar to the encoder. The only difference is at the beginnind of each layer D1, D2, D3, ..., where we use the upsampling layer to increase the size, instead of the convolutional layer. The decoder contains 5 upsampling layers and 11 convolutional layers (Fig. 6.5). The loss function is the mean square error (MSE) to measure the difference between the gated images and the neural network output. The Adam optimizer is used for convergence. Batch size is 50. The number of epochs is 100. We use one GPU node of the cluster at Florida Tech. It has 4 Nvidia Tesla K40m GPUs. And we utilize all 4 GPUs to train the model by Keras.

It took 151570 seconds for training (around 42 hours). The MSE of the test set is 0.005478. Fig. 6.6 demonstrates an example of the results. The top row highlights the original myocardium image. The middle row presents the reconstructed output from CED and the final row displays the reconstruction error. As can be observed, the reconstruction seemed to be smoothed by the neural network. Notice that under each image in the second row, we also put the MSE, structural similarity index measure (SSIM), and peak signal-to-noise ratio (PSNR) to mathematically show the difference

between the produced images and the gated images. From those values, we noticed that even the MSE is relatively low, but the SSIM and PSNR are not that good. Recall that the range of SSIM is from 0 to 1 and when two images are very similar, the SSIM will be close to 1. On the other hand, higher PSNR value means the better quality of the reconstruction. PSNR around 20 (dB) means the quality is acceptable, but humans can easily see the difference by eyes. These statistical values are not beyond our anticipation. Because it's obvious that the reconstruction from the neural network output is cleaner and smoother than the original gated image. However, we need to peform more extensive analysis to measure whether this kind of smoothing makes the reconstruction more accurate or not. We have however demonstrated in this work, the CED model does provide potential as an efficient mechanism to not only reconstruct PET images but also provide a data correction methodology which requires exploration.

6.2 Introduction of zero-shot reconstruction

In this work we investigate the effectiveness of transfer learning for cardiac gated PET (positron emission tomography) image reconstruction in absence of any training data from in-vivo imaging. For these purposes we utilize computational phantom imaging data of a *Mouse Whole Body* (MOBY) to train a deep learning network that is capable of reconstructing in-vivo PET sinograms, while simultaneously removing motion artifacts, i.e., producing a gated output. In this sense, our work is related to *zero-shot learning*. Our model, whilst trained on phantom data, uses in-vivo ungated sinogram as the input and reconstructs gated or motion-corrected images as the output. The major significance of our work is that a gated reconstruction may be produced, void of motion artifact without the need of using gating hardware or the requirement for in-vivo images of training data. We utilize an innovative convolutional encoder-decoder

(CED) artificial neural network (ANN) architecture that is enhanced with so called self-attention layers (CEDA). The self-attention layers focuses on the cardiac region of interest (ROI), and thus, is capable of reconstructing PET images of a motion free heart. We demonstrate our model's superior performance in comparison to the conventional iterative maximum likelihood expectation maximization (MLEM) algorithm in recognizing healthy versus diseased myocardium.

Medical image reconstruction is a complex process which attempts to solve the inverse problem of image formation, traditionally using physical assumptions on count statistics and the relation between the object and the measurements during acquisition. Multiple sources of error can corrupt these assumptions and obfuscate the reconstruction process. The ANN has been shown to have the capability to learn the measurement model transparently and reconstruct images without any explicit modeling of the involved parameters [87, 88, 93, 94, 133, 134]. However, the primary drawback of ANN-based modeling is the requirement for a large amount of training data a priori, which is often unrealistic in the medical imaging field. In this paper we demonstrate that one may artificially generate such training data with prior knowledge. We still need to model the system manually in the context of forward modeling or training data generation, but that is much easier than solving the inverse problem.

In this research, we artificially generate mouse cardiac images with motion by enhancing MOBY phantom, forward project them for a specific PET data acquisition process, and then, train a proposed ANN architecture with these datasets. We show that the trained model performs better than the conventional iterative reconstruction method. To address the issue of cardiac motion, one of the most important extant techniques is to use external *gating* devices. It uses electrocardiogram (ECG) to record the cardiac motion signal [135, 136] and temporally bins data from pre-determined cardiac phases. Subsequently, image reconstruction is performed on projection data

from each phase, thus, producing motion compensated images [135–138]. Also, literature is abundant with motion-compensation algorithms, including some usage of ANN as a post-processing step to reduce motion-blur [139, 140]. Our primary contribution is that the proposed method here obviates any need of hardware devises or motion-compensation software, even for the purpose of training a machine learning algorithm.

6.3 Methodology

6.3.1 Data

The problem addressed in this work is a motion compensated pre-clinical cardiac PET study using the motion corrupted sinogram data using an ANN model trained with only simulated phantom data. This necessitates generating appropriate object and motion model in simulation. We enhanced the 4D MOBY [141,142] phantom for ANN training purposes by creating a pseudo pre-clinical cardiac PET series through pre-processing steps described below. However, MOBY can only display the shapes and positions of the tissues and organs. We designed a special pre-processing step to approximate the real data to the maximum extent.

6.3.1.1 Training data generation with MOBY

Motion simulation: The MOBY phantom is a 3D mouse model with the ability to generate realistic voxel based objects at different phases of cardiac and respiratory motion [143–145]. The MOBY phantom is designed from underlying high resolution gated multi-detector CT (MDCT) data [7, 146]. It contains in total one hundred time frames over a complete cardiac cycle. Fig. 6.7 shows the change of volume in the four heart chambers of heart over time. The MOBY data we created included a complete

cardiac cycle. The final generated MOBY phantom object consisted of 10 output frames, which were evenly distributed over the cardiac cycle. Say the purpose first, I assume to generate PET images, we changed the myocardium tissue intensity and the heart size to represent 2793 different MOBY 3D phantoms. For computational efficiency we selected four 2D slices per 3D mouse phantom which spanned the volume of the myocardium for limitation in computational resources. Two data series are constructed, (a) a motion corrupted data series, which consists of the summation of all 10 cardiac phases and hence is corrupted by motion blur and (b) a single end-diastolic phase, which represents a gated frame void of motion blur. Thus, in total four gated slices and four motion corrupted slices of pseudo PET data were generated from the MOBY object data. In total, altogether $4 \times 2793 = 11172$ slices (both gated or noisy types) were utilized for training.

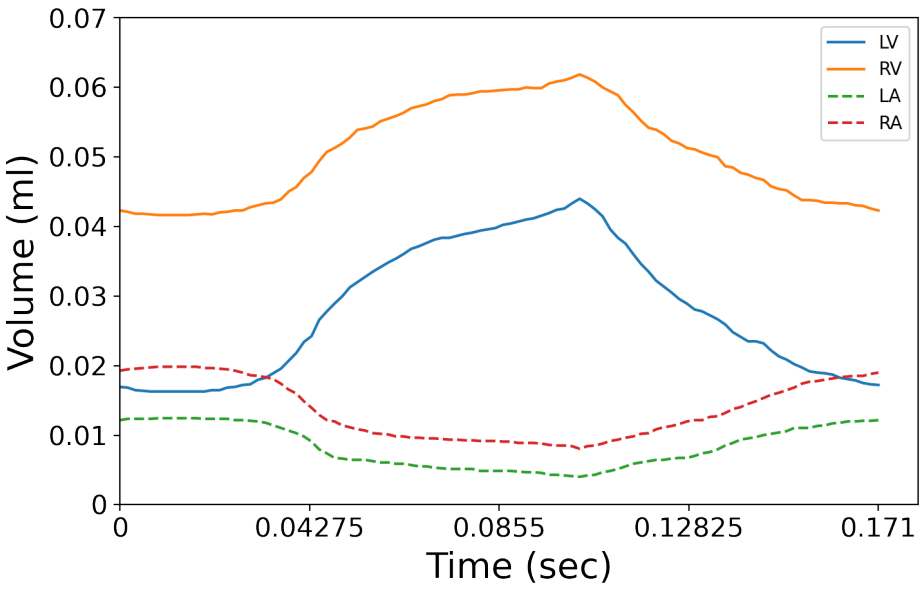


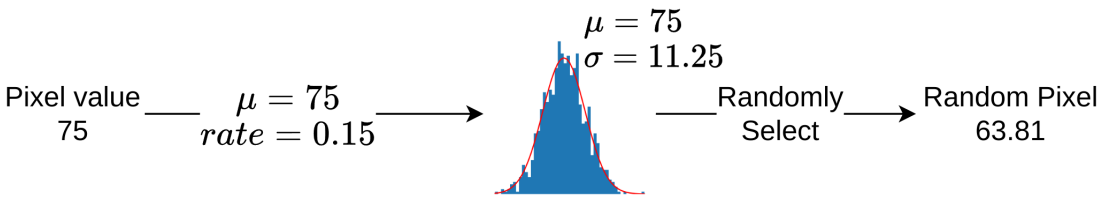
Figure 6.7: The motion curves of MOBY showing the volume changes of the heart chambers over time [7].

Pixel randomization: The MOBY phantom is a digitized object wherby every voxel in each organ is assigned the same voxel value. To generate realistic PET data from the

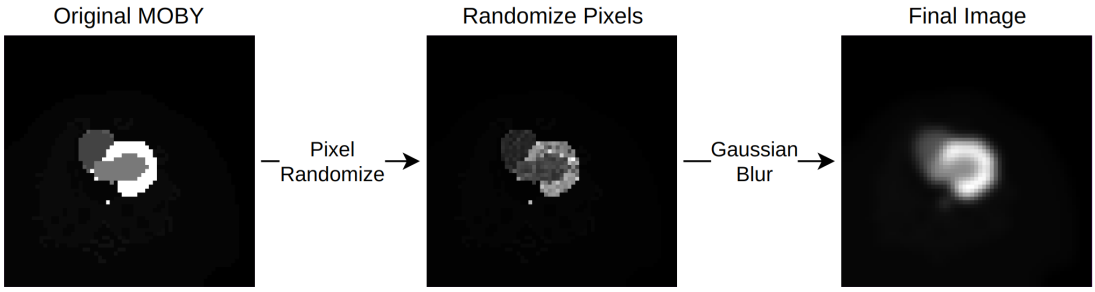
phantom object we use a voxel variation method to randomize the voxels such that they are more representative of count statistics observed in a PET image. In a real PET scan, it is impossible that all the pixel values are the same within the same tissue or organ. For this reason, we design a pixel variation method to randomize the pixel values. Every pixel within the same region (e.g., in the left ventricle of myocardium) is re-assigned to a random value that is generated by a Gaussian distribution. The Gaussian distribution is with a mean μ that is the scaled original MOBY pixel value and a chosen standard deviation (stdv) δ. We varied the scaling factor r and the δ to create multiple images from each phantom slice. A range of r was arbitrarily selected from 0.1 to 0.3, in increments of 0.01. For example, suppose a pixel has the value 75 in the original MOBY image, then $\mu = 75$. If the random rate r is 0.15, then $\delta = 0.15 \times 75 = 11.25$, and the Gaussian distribution will be with $\mu = 75$ and $\delta = 11.25$. Next, the new pixel value is picked up randomly from this distribution. Fig. 6.8a illustrates the process of this pixel randomization method. Gaussian convolution is applied to the result of the pixel randomization to alleviate the well defined edges that are observed in the MOBY phantom; thus generating a more realistic PET image. Hence, we further applied a Gaussian kernel over each image to blur the edges between different regions. In Fig. 6.8b, we can see how the MOBY image changed after randomizing the pixel and blurring the edges.

Training data augmentation: For increased performance of the ANN model image augmentation of all 11172 slices was performed. The augmentation step consisted of eight affine transformations, including shearing, scaling, rotation, translation, and flipping. The ranges of the parameters of those transformations are listed below:

- Shearing: 0 to 0.4, increment 0.1.
- Scaling: 8 to 10, increment 1.



(a) This is an example to show the pixel randomization. The mean of the Gaussian distribution is the same as the pixel value, while the stdv is the product of a random rate and the mean (here the rate is 0.15, thus stdv is $75 \times 0.15 = 11.25$).



(b) What the final MOBY image looks like after pixel randomizing and Gaussian filtering.

Figure 6.8: The two steps of the MOBY data post-processing: Pixel randomization and Gaussian blur. With these two steps, the original MOBY image is rendered more realistic.

- Rotation: 0 to 350 in degree, increment 10.
- Translation: -16 to 16, increment 1.
- Flipping: 0 to 4, increment 1, which means "vertical flipping", "horizontal", "both vertical and horizontal", and "no flipping" respectively.

For each affine transformation of a 2D slice, the affine parameters were randomly selected. Augmentation results in $11172 \times 8 = 89376$ images for both the gated and motion corrupted simulated PET data. To create matched sinograms for the synthesized PET data forward projection was performed. The acquisition parameters of the forward projection were selected to match the PET system in which the testing data was acquired. A system matrix modeled the image acquisition process to convert sinogram to an image. Following image augmentation and forward projection, 89376 images and sinograms were created. As a result, we have the correct noisy sinogram

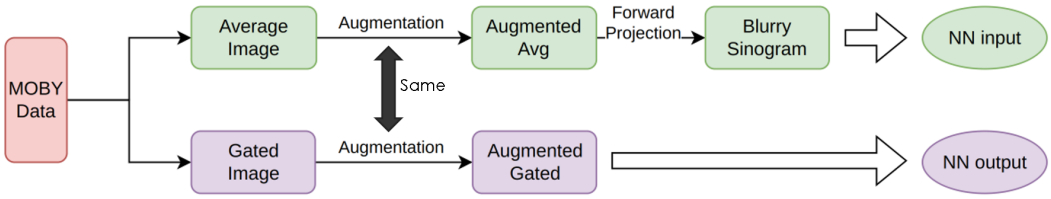


Figure 6.9: The workflow of creating the training data from MOBY.

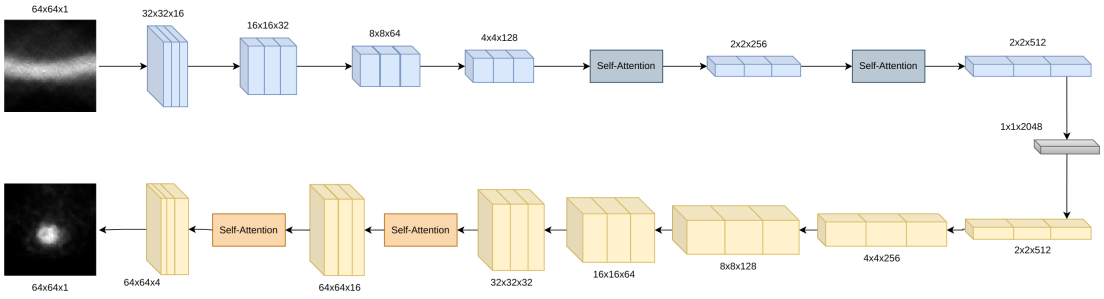


Figure 6.10: Convolutional encoder-decoder with self-attention. Each small block contains one convolutional layer followed by a batch normalization layer and a Leaky ReLU layer. In the encoder side (upper row), we set convolutional stride to 2 to downsampling the input features. In the decoder side (lower row), we used upsampling layer to upsample the input features.

that corresponds to the gated image used in training the ANN as the respective input and output pair, respectively. Fig. 6.9 shows the process of generation of training data from MOBY phantom.

6.3.1.2 Test data from pre-clinical PET

For testing, the mouse study was conducted at PET Imaging Centre (PETIC), Cardiff University, UK, using a Mediso nanoScan 122S small bore PET/CT imaging system (Mediso Medical Imaging Systems, Budapest, Hungary). The protocol was standardised to ensure optimal and consistent biodistribution of ^{18}F-FDG. Briefly, mice had food withdrawn and were warmed at 37 °C for 1 hour before scanning, which was carried out under isoflurane-anaesthesia (1.5-2% in 1 L/min oxygen). An intraperitoneal injection of 100–150 μL of Iohexol CT contrast agent (647 mg/mL) (Omnipaque 300,

GE Healthcare Inc., Marlborough, MA, USA) was followed by a tail vein injection of 32 ± 8 MBq ^{18}F-FDG. Mice were maintained under anaesthesia for 50 minutes post injection to allow the uptake of ^{18}F-FDG before a 20-min cardiac gated PET scan was acquired followed by a 2.5 min whole-body CT scan. We used retrospective pre-clinical cardiac PET data over ten mice, three of which were healthy and the rest were from obese and diabetic models. Since we focus on the heart, approximately 10 slices around the heart were picked up for each mouse (the exact number depends on the respective mouse's image). However, due to the computer memory concern, we reduced the dimension by cropping around heart to 64×64 pixels (original dimension of different mice is around 100×100). To reduce the dimension of the sinogram, we first back project the original sinogram, crop the backprojection to 64×64 pixels, and then forward project it to get the new cropped sinogram (using the *system matrix* with 64 number of projections). Finally, we used ninety-nine 2D test images for validation purposes.

6.3.2 ANN model

We used the self-attention components while adapting the original CED architecture proposed in the literature. The conventional CED [94] contains two parts: an encoder to extract the features from the sinograms (downsampling) which works like a normal convolutional neural network (CNN), and a decoder to restore the image from the encodings which are the outputs of the encoder (upsampling). In each of the encoder and decoder, there are 6 modules. Each module has 3 convolutional blocks (CB). Every CB consists of one convolutional layer, a batch normalization layer, and a leaky ReLU layer. Since the encoder is performing downsampling and the decoder is performing upsampling, the CBs are slightly different in the module of the encoder and decoder. In the module of the encoder, the first CB has a stride 2 convolutional layer while the remaining two CBs are with stride 1. However, in the module of the decoder, the first

CB is using a convolutional transpose layer with stride 2, and rest of the two CBs are using the normal convolutional layer with stride 1. We developed this architecture by trial and error based on previously presented work [94].

The self-attention [66, 68] component was inspired from the attention mechanism [66]. It attempts to to simulate the human visual attention, which focuses on some areas of an image instead of the overall image. In deep learning, attention mechanism will give a score to interpret the importance of this element (a pixel in a image, or a word in a sentence). The self-attention component will take the output x from the previous layer and derive the key (K), value (V), and query (Q) by using three 1×1 filter convolutional layers. Next, we use softmax on the dot product of the key-query pair to get the attention map (a matrix with the score of each pixel), and perform the dot-product again with V to generate the self-attention feature map O. At last, the output O of the self-attention layer is scaled by an arbitrarily chosen parameter λ and added back to the original input s. Because of this, our ANN model was able to capture the spatial relationship between different regions. Since we add the self-attention component into the encoder and decoder of the CED, we name our proposed architecture as CEDA.

6.3.3 Statistical Analysis

As a reminder, we used simulation data (MOBY) to train our model and real data (pre-clinical gated PET study) to test it. Two types of reconstructioned images are used in our work. (a) Gated *ordered set expectation maximization* (OSEM is an efficient version of the MLEM algorithm used by the imaging system vendors, Mediso, Hungary) reconstructed images from the PET machines internal software at Cardiff University; and, (b) reconstructions from our trained CEDA model. All the reconstructed images were 64×64 (with the pixel size 0.4×0.4 mm^2). The gated OSEM reconstructions at end

diastole were selected as the reference image (gated ground truth). The reconstructed images from CEDA were compared to the reference image, in order to measure the quality of the reconstruction. We have used three types of statistical measures for comparison that are described below.

6.3.3.1 Visual information fidelity

Images are often characterized or measured by traditional statistical models. However, these approaches sometimes produce a different conclusion compared to the subjective assessment of human visual system (HVS). Visual information fidelity (VIF) can quantify the information which the human brain perceives from the reference images. This allows quantification of the HVS and is commonly used in image distortion models [129]. When measuring the Laboratory for Image & Video Engineering (LIVE) image quality assessment database, the Spearman Rank-Order Correlation Coefficient (SROCC) between the VIF scores of distorted images and the corresponding human opinion scores is 0.96 [130], the VIF scores proved to be very close to the human assessment of image quality. The equation is the same as we showed in Chapter 5 (Eq. 5.1). VIF is usually between $[0, 1]$. For the tested image being a copy of reference image without any information lost, $VIF = 1$. Note that if only the linear contrast of the reference image is enhanced without any increase of the noise, VIF may be greater than 1.

6.3.3.2 Signal-to-noise ratio and Image contrast

To calculate the signal-to-noise ratio (SNR) and the image contrast ratio (CR), we defined the ROI as a rectangular box around the heart, thus, ignoring the larger background. The SNR is computed by the ratio of the mean intensity of the ROI and the standard deviation (stdv) of the background, while CR is the mean of ROI divided by

the mean of the background.

6.4 Results

Our test input is the non-gated sinogram of mouse, and the output is a CEDA reconstructed motion free image, which is compared to the OSEM gated output. Fig. 6.11 shows some sample reconstructions from the gated OSEM reconstructions (first row) and from our CEDA model (second row) trained with motion-enhanced MOBY phantoms (Fig. 6.11). With 200 epochs, the training time was 24488.31 sec and the average evaluation time (reconstruction) per 2D slice was 0.0011 sec. The machine to train this model has Intel Core i7-9700K 3.6 GHz 8-Core Processor, 1 TB Solid State Drive, NVIDIA Titan Xp 12 GB Video Card, 2×16 GB DDR4 Memory.

Fig. 6.12 shows the comparisons with VIF, SNR and CR for all the testing images. For all VIF, SNR, and CR, a higher score means better performance. A higher VIF indicates that the visual information being presented or processed in the reconstructed image is more faithful, accurate, and closer to the reference image. A high SNR shows a higher quality and clarity of the reconstructed image, with less interference from noise.

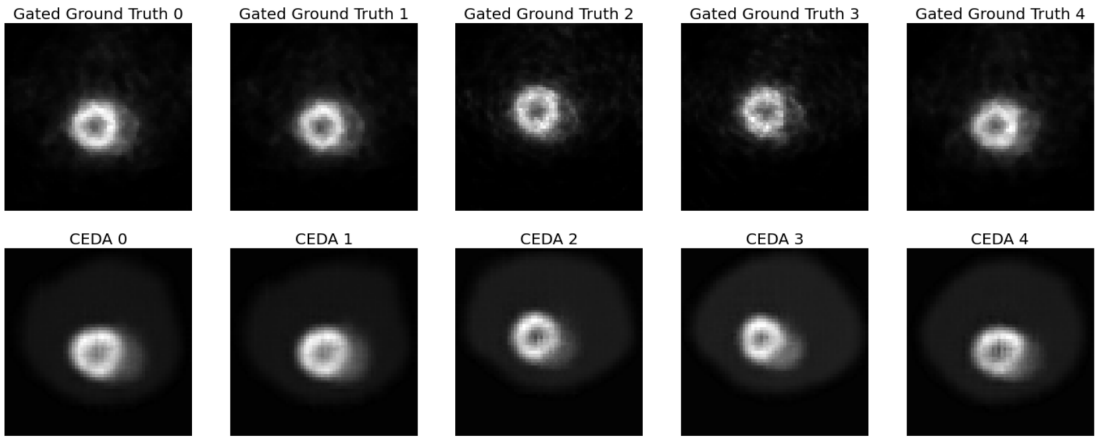


Figure 6.11: Sample reconstructed images. First row: gated OSEM. Second row: CEDA. Third row: Non-gated MLEM.

A higher CR makes the reconstructed image easier to distinguish and differentiate between different anatomical features or regions of interest.

For SNR, all the CEDA reconstructed images have higher SNR than those from MLEM and the gated OSEM (Fig. 6.12b). On the average, SNR of the CEDA was 16.74 ± 1.51, the MLEM was 6.59 ± 2.25, and the OSEM was 7.82 ± 1.57 (Fig. 6.13b). The CR showed that CEDA also achieved higher values for all the testing images (Fig. 6.12c). The mean of CEDA CR was 24.17 ± 2.63, that of MLEM was 12.56 ± 3.98, and that of the gold standard OSEM was 12.03 ± 3.22 (Fig. 6.13c).

6.5 Discussion

The results presented above shows that the proposed CEDA model can reconstruct the motion-free heart image of the real mouse from the non-gated sinogram with reasonable performance. A primary novelty of the work is that our model was trained by the synthetic data only without using any real data. Compared to MLEM reconstruction from the same non-gated sinogram data, our model obtained higher VIF scores (Fig. 6.13a), which indicated that the CEDA model produces motion-compensated images with better information content compared to that with MLEM (with respect to the gated OSEM as the ground truth).

A noteworthy observation is that our model has significantly higher SNR for the reconstructions. The mean SNR increased 113.96% compared to the gated OSEM, respectively. Our model also achieved superior CR. The relative rates of increase were 100.96% compared to the gated OSEM. Also, visual inspections of reconstructed images showed that CEDA can automatically emphasize the heart, while decreasing the noise level.

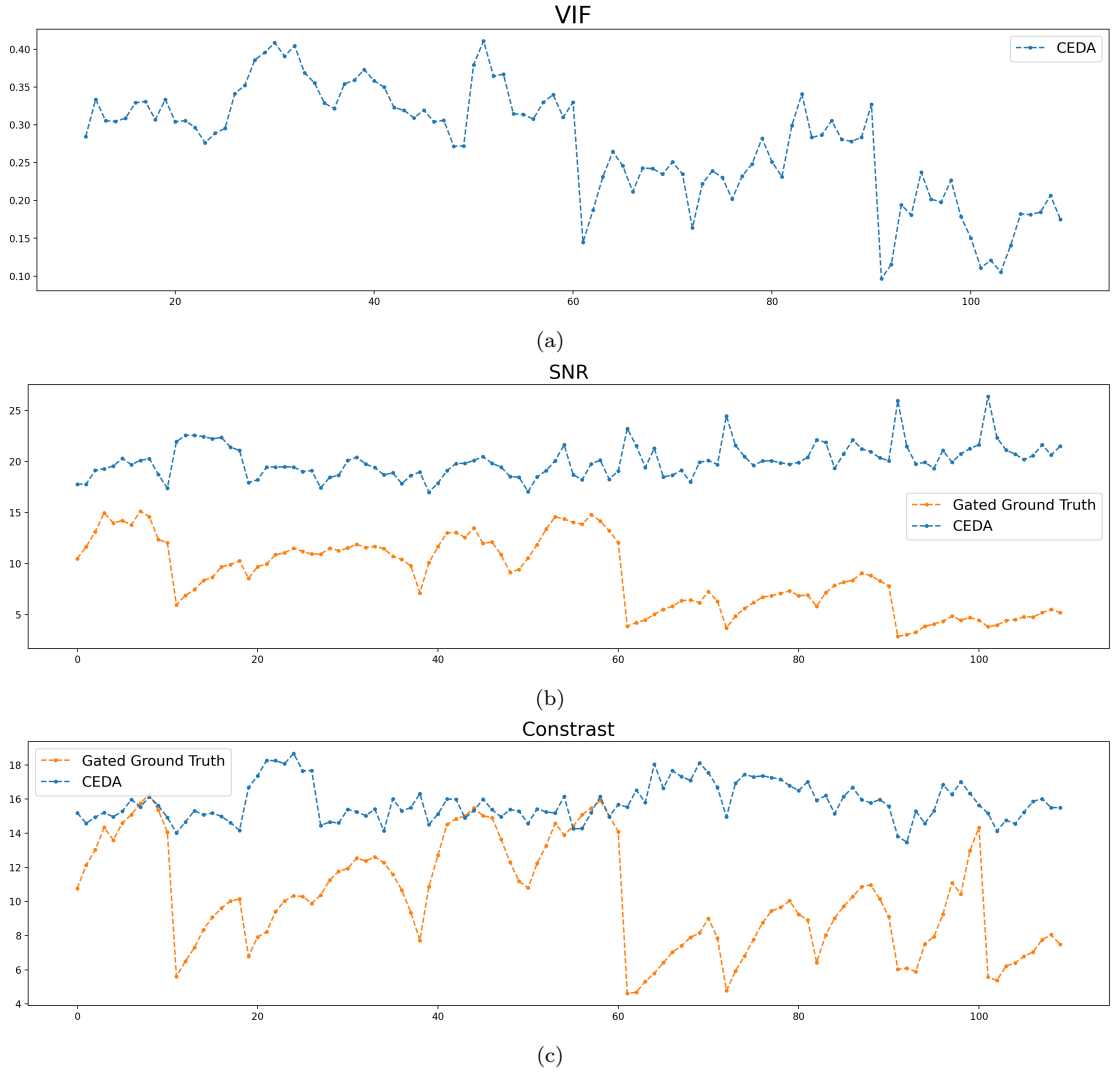
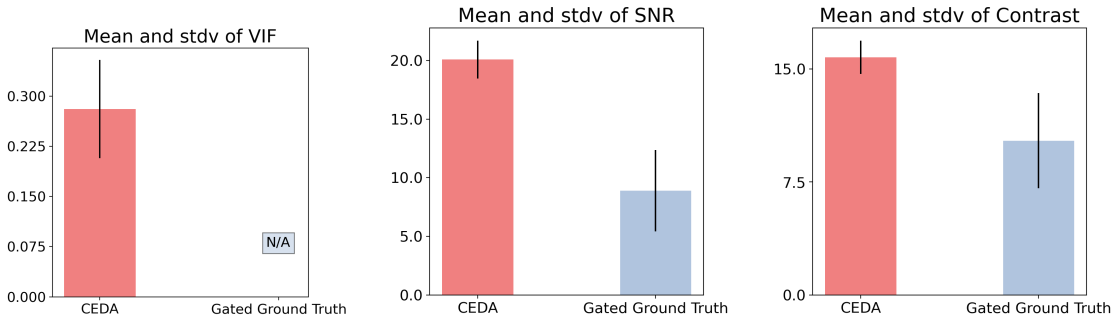


Figure 6.12: Measuring the performance of CEDA using VIF, SNR, and CR. Note that to compute VIF, the gold standard OSEM reconstructions were used as reference images, while SNR and CR are computed for each image independently. Hence, in (a) there are two curves and in (b) and (c) there are three curves. In each plot, X-axis represents the arbitrarily ordered indices of test images, and the Y-axis represents the values of the corresponding measurements.

6.6 Conclusion and Future Works

In this paper, we demonstrated that the proposed CEDA model can reconstruct a gated image given its non-gated sinogram of pre-clinical PET imaging, while the model was trained only with the phantom data. The significance of the work is that it opens the possibility for training ANN models without the availability of real data using TL from



(a) (b) (c)

Figure 6.13: The mean and stdv of VIF, SNR, CNR from Fig. 6.12.

realistic simulation. This shows that hardware gating mechanisms are not required at all to reconstruct gated images, provided simulation of cardiac phases are modeled appropriately. This is achieved with three innovations: (1) creating a more realistic phantom by enhancing the MOBY phantom with randomization of pixel values, (2) cardiac motion modeling, and (3) proposing an ANN architecture called CEDA to reconstruct the gated image from non-gated sinogram. For each experiment we have used VIF, SNR, and image contrast for comparison against conventional MLEM reconstruction, and have shown that CEDA performs better. We also have shown that the proposed model has better capability to detect diseased heart (obese mice, in our case) than that by the MLEM reconstruction.

One of the limitations of the current study comes from MOBY phantom generation. The MOBY data we used did not contain defects. Also, even though we used augmentation to vary the heart shape, we couldn't produce independent variation of the shapes of different parts of a heart. The augmentation process can only alter the size of the whole heart, and the shape of each part changes proportionately. This means our augmented hearts may not represent different realistic cardiac geometries and conditions. Future work will take these issues into consideration and create a more realistic phantom. This should finally improve the reconstruction quality when testing

more diverse real data.

Another direction of this work will involve directly reconstructing fully 3D data that are computationally challenging both at clinical and research settings. It is quite reasonable to expect that the ANN model will be able to learn even better reconstruction in 3D than in 2D. Also, our image and motion modeling can be further improved toward real scenarios, e.g., by adding respiratory motion model. Especially, for human cardiac imaging, respiratory modeling is important. Finally, we would like to train our model for direct recognition of different cardiac diseases than just the obesity (which possibly appeared as cardiac hypertrophy [147]) with improved simulation for training, as we have mentioned above.

Chapter 7

Conclusion and future work

In this dissertation, We explore and explain some traditional problems in medical imaging, as well as basic concepts and applications of artificial neural networks. We also showcase some research achievements that combine neural networks and medical imaging. Following, we present our achievements of applying neural networks to solve medical imaging problems.

In chapter 4, we demonstrate the use of ANN to solve some of the simple medical imaging problems, such as FFT prediction, attenuation coefficient estimation of a uniform disk, U-shape mask image reconstruction, motion function recovery and motion elimination. The results of these experiments indicate that ANN has sufficient ability to address the medical imaging problems. The success of these preliminary experiments laid the foundation for subsequent experiments. Additionally, we use different ANN models to solve the problems in these works, like FCN, CNN, and CED. The problem of FCN is that FCN can solve the issue but require much more resources than other ANN models. A classic CNN can effectively extract desired features, including parameters of the object like the position and attenuation coefficient, but it cannot independently perform the reconstruction task. When we use CED, it can perfectly extract features

from blurry sinograms and reconstruct the clear images. As a consequence, our main tasks are based on CED. However, the limitations of these experiments are also evident, as the images and parameters used were too idealized. Therefore, the following experiments were improved in two ways: (1) more realistic data or even real data, and (2) improve CED so that it can have better performance on a more sophisticated scenario.

In chapter 5, we used another simulation data to simulate the heart. Compared to the U-shape mask image, this pseudo heart image is closer to the real myocardium with introduced tissue type separation including infarction. Further, the affine transformation including translation, rotation, and scaling is applied to imitate the cardiac motion. In order to improve the performance, we adopt the self-attention mechanism to modify CED. We call it CEDA. Further, we also utilize CEDA to reconstruct the motion-free image from the noisy human data. To measure the performance, we introduce VIF to compare the performance between CEDA and the original CED. The result show that CEDA has a good quality on the reconstruction with motion correction from the motion blurred Radon transformed image. It learns the data acquisition (or underlying physics) model and the motion-model from a training set, and does not need any additional hardware (e.g., gating) or software (e.g., system-matrix generation).

In chapter 6, we advance our research further. The previous experiments prove that the ANN model can learn the underlying physics and the motion model of the data. As a result, we abandon the traditional approach of using the same type of data to train the ANN model. Instead, the mouse phantom data MOBY is employed to train CEDA and we test it using real mouse data (pre-clinical PET imaging). In this work, we: (1) enhance the MOBY phantom by randomizing pixel values to create a more realistic phantom, (2) model cardiac motion, and (3) propose a new ANN architecture called CEDA, which is used to reconstruct gated images from non-gated sinograms.

Moving forward, there are several avenues for future research that could build upon our work. One of the future directions is to extend this work by directly training and validating the model with 3D images instead of 2D slices as done in this study. However, using 3D images presents significant challenges. Firstly, the computational resources required for training with 3D images are currently impractical, both in research and clinical settings. Secondly, the amount of training and validation data needed for 3D models is considerably higher than what is commonly available from clinics. Therefore, as is standard practice in the literature, we sliced the 3D images around the region of interest to obtain sufficient 2D data sets for this study. In the future, we intend to collect more 3D data to improve the model's training and to validate its motion correction capability in detecting cardiac infarction. Moreover, we plan to apply this technique in non-medical areas, such as astrophysics, where motion can also affect imaging.

There is also a limitation for the experiment of reconstructing real data using ANN model that is trained by the phantom data. Thus, the present study is related to the generation of the MOBY phantom. Specifically, the MOBY data utilized in this study did not include any defects. Moreover, although we employed augmentation techniques to increase the variability of the heart shapes, we were unable to produce independent variations in the shapes of different parts of the heart. Our augmentation process only enabled alteration of the overall heart size, with each part changing proportionately. As a result, our augmented hearts may not accurately represent various realistic cardiac geometries and conditions. Moving forward, future studies will need to address these limitations by creating a more realistic phantom to improve the reconstruction quality when testing with more diverse real data.

Furthermore, our work aims to enhance the signal intensity and improve the overall image quality, thereby unlocking the untapped potential of our research in this do-

main. In recent years, there have been notable advancements in this area, with studies focusing on super-resolution techniques for SPECT reconstruction [148–150], as well as cross-modality synthesis, such as converting CT images to MRI [151–153]. These studies demonstrate the effectiveness of deep learning approaches in improving image resolution and quality. The studies mentioned before inspire us to explore further advancements in SPECT reconstruction. Notably, SPECT is more widely adopted in clinical practice compared to PET due to its significantly lower cost. While PET offers superior resolution, it is hindered by inherent system errors resulting from the emitted positron transmission. In other words, if SPECT can achieve higher resolution, it would mark a revolutionary breakthrough benefiting both hospitals and patients alike. By bridging the resolution gap between SPECT and PET, we can enhance diagnostic capabilities and provide more accurate and detailed imaging results.

Bibliography

[1] L. A. Shepp and B. F. Logan, "The fourier reconstruction of a head section," *IEEE Transactions on Nuclear Science*, vol. 21, no. 3, pp. 21–43, 1974.

[2] N. M. Abbasi, "Note on radon and iradon transforms and matlab's iradon on the all-at-once call vs. the one-at-time call," https://www.12000.org/my_notes/note_on_radon/index.htm, 2008, accessed: 2018-10-17.

[3] B. F. Hayden, "Slice reconstruction," http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/AV0405/HAYDEN/Slice_Reconstruction.html, 2005, accessed: 2018-10-17.

[4] C. Berger, "Perceptron - the most basic form of a neural network," https://appliedgo.net/perceptron/, 2016, [Online; accessed 15-October-2018].

[5] C. Coulombe, "The revenge of perceptron -learning xor with tensorflow." https://medium.com/@claude.coulombe/the-revenge-of-perceptron-learning-xor-with-tensorflow-eb52cbdf6c60, 2017, [Online; accessed 17-October-2018].

[6] A. Deshpande, "A beginner's guide to understanding convolutional neural networks," https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/, 2016, [Online; accessed 16-October-2018].

[7] W. P. Segars, K. Taguchi, G. S. K. Fung, E. K. F. M.D., and B. M. W. Tsui, "Effect of heart rate on CT angiography using the enhanced cardiac model of the 4D NCAT," in *Medical Imaging 2006: Physics of Medical Imaging*, M. J. Flynn and J. Hsieh, Eds., vol. 6142, International Society for Optics and Photonics. SPIE, 2006, p. 61420I.

[8] Apollo General Physician, "What is a SPECT scan commonly used for?" 2021, [Online; accessed January 1, 2021]. [Online]. Available: https://healthlibrary.askapollo.com/what-is-a-spect-scan-commonly-used-for/

[9] P. P. Bruyant, "Analytic and iterative reconstruction algorithms in spect," *Journal of Nuclear Medicine*, vol. 43, no. 10, pp. 1343–1358, 2002. [Online]. Available: https://jnm.snmjournals.org/content/43/10/1343

[10] National Health Service, "PET scan," [Online; Page last reviewed: 17 March 2021]. [Online]. Available: https://www.nhs.uk/conditions/pet-scan/

[11] J. Radon, "1.1 über die bestimmung von funktionen durch ihre integralwerte längs gewisser mannigfaltigkeiten," *Classic papers in modern diagnostic radiology*, vol. 5, p. 21, 2005.

[12] N. Megherbi, T. P. Breckon, G. T. Flitton, and A. Mouton, "Radon transform based automatic metal artefacts generation for 3d threat image projection," in *Optics and Photonics for Counterterrorism, Crime Fighting and Defence IX; and Optical Materials and Biomaterials in Security and Defence Systems Technology X*, vol. 8901. International Society for Optics and Photonics, 2013, p. 89010B.

[13] G. T. Herman, *Fundamentals of computerized tomography: image reconstruction from projections*. Springer Science & Business Media, 2009.

[14] J. Frank, *Three-dimensional electron microscopy of macromolecular assemblies: visualization of biological molecules in their native state*. Oxford University Press, 2006.

[15] D. E. Dudgeon and R. M. Mersereau, *Multidimensional Digital Signal Processing Prentice-Hall Signal Processing Series*. Prentice-Hall, Englewood Cliffs, NJ, 1984.

[16] T. F. Budinger, G. T. Gullberg, and R. H. Huesman, "Emission computed tomography," in *Image reconstruction from projections. Implementaton and applications*, 1979.

[17] K. Lange, R. Carson *et al.*, "Em reconstruction algorithms for emission and transmission tomography," *J Comput Assist Tomogr*, vol. 8, no. 2, pp. 306–16, 1984.

[18] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[19] Y. Freund and R. E. Schapire, "Large margin classification using the perceptron algorithm," *Machine learning*, vol. 37, no. 3, pp. 277–296, 1999.

[20] F. Rosenblatt, *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory, 1957.

[21] M. Minsky and S. A. Papert, *Perceptrons: An introduction to computational geometry*. MIT press, 2017.

[22] F. Rosenblatt, "Principles of neurodynamics. perceptrons and the theory of brain mechanisms," CORNELL AERONAUTICAL LAB INC BUFFALO NY, Tech. Rep., 1961.

[23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, p. 533, 1986.

[24] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[25] Y. LeCun *et al.*, "Lenet-5, convolutional neural networks," *URL: http://yann.lecun.com/exdb/lenet*, p. 20, 2015.

[26] K. Hayat, "Super-resolution via deep learning," *arXiv preprint arXiv:1706.09077*, Jan 2017. [Online]. Available: https://arxiv.org/abs/1706.09077

[27] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *2011 international conference on computer vision*. IEEE, 2011, pp. 2018–2025.

[28] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[29] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, 2016. [Online]. Available: http://distill.pub/2016/deconv-checkerboard

[30] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, 2010, pp. 2528–2535.

[31] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," *arXiv preprint arXiv:1603.07285*, 2016.

[32] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 12 1989. [Online]. Available: https://doi.org/10.1162/neco.1989.1.4.541

[33] Y. LeCun *et al.*, "Generalization and network design strategies," *Connectionism in perspective*, vol. 19, no. 143-155, p. 18, 1989.

[34] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a back-propagation network," *Advances in neural information processing systems*, vol. 2, 1989.

[35] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[36] K. Chellapilla, S. Puri, and P. Simard, "High performance convolutional neural networks for document processing," in *Tenth international workshop on frontiers in handwriting recognition*. Suvisoft, 2006.

[37] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in *Twenty-second international joint conference on artificial intelligence*, 2011.

[38] OFFICIAL IJCNN2011 COMPETITION, "IJCNN 2011 competition result table," [Online; Last modified on May 10.]. [Online]. Available: https://benchmark.ini.rub.de/?section=gtsrb&subsection=results

[39] J. Schmidhuber, "History of computer vision contests won by deep cnns on gpu," *AI Blog*, 2017.

[40] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[41] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[42] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3642–3649.

[43] M. Matsugu, K. Mori, Y. Mitari, and Y. Kaneda, "Subject independent facial expression recognition with robust face detection using a convolutional neural network," *Neural Networks*, vol. 16, no. 5, pp. 555–559, 2003, advances in Neural Networks Research: IJCNN '03.

[44] M. T. Review, "The face detection algorithm set to revolutionize image search."

[45] S. S. Farfade, M. Saberian, and L.-J. Li, "Multi-view face detection using deep convolutional neural networks," Nov 2015, in International Conference on Multimedia Retrieval 2015 (ICMR). [Online]. Available: https://arxiv.org/abs/1502.02766

[46] C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[47] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of cnn and rnn for natural language processing," 2017. [Online]. Available: https://arxiv.org/abs/1702.01923

[48] W. Wang and J. Gang, "Application of convolutional neural network in natural language processing," in *2018 International Conference on Information Systems and Computer Aided Education (ICISCAE)*, 2018, pp. 64–70.

[49] A. M. Alayba, V. Palade, M. England, and R. Iqbal, "A combined cnn and lstm model for arabic sentiment analysis," in *Machine Learning and Knowledge Extraction*, A. Holzinger, P. Kieseberg, A. M. Tjoa, and E. Weippl, Eds. Cham: Springer International Publishing, 2018, pp. 179–191.

[50] I. Wallach, D. Michael, and A. Heifets, "Atomnet: A deep convolutional neural network for bioactivity prediction in structure-based drug discovery," Oct 2015, arXiv preprint Machine Learning. [Online]. Available: https://arxiv.org/abs/1510.02855

[51] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," Jun 2015, appear at ICML Deep Learning Workshop 2015. [Online]. Available: https://arxiv.org/abs/1506.06579

[52] K. Chellapilla and D. Fogel, "Evolving neural networks to play checkers without relying on expert knowledge," *IEEE Transactions on Neural Networks*, vol. 10, no. 6, pp. 1382–1391, 1999.

[53] ——, "Evolving an expert checkers playing program without using human expertise," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 4, pp. 422–428, 2001.

[54] D. B. Fogel, *Blondie24: Playing at the Edge of AI*. Morgan Kaufmann, 2002.

[55] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, p. 234–241, 2015. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-24574-4_28

[56] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2017.

[57] B. H. Menze *et al.*, "The multimodal brain tumor image segmentation benchmark (BRATS)," *IEEE Transactions on Medical Imaging*, vol. 34, no. 10, pp. 1993–2024, 2015.

[58] S. Bakas *et al.*, "Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features," *Scientific Data*, vol. 4, no. 1, pp. 1–13.

[59] ——, "Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge," Apr 2019, the International Multimodal Brain Tumor Segmentation (BraTS) Challenge. [Online]. Available: https://arxiv.org/abs/1811.02629v3

[60] B. Van Ginneken, T. Heimann, and M. Styner, "3d segmentation in the clinic: A grand challenge," in *MICCAI workshop on 3D segmentation in the clinic: a grand challenge*, vol. 1, 2007, pp. 7–15.

[61] "Segmentation of the liver competition 2007 (SLIVER07)," https://sliver07.grand-challenge.org/, accessed: 2017-10-10.

[62] T. Heimann *et al.*, "Comparison and evaluation of methods for liver segmentation from CT datasets," *IEEE Transactions on Medical Imaging*, vol. 28, no. 8, pp. 1251–1265, 2009.

[63] F. Nazem, F. Ghasemi, A. Fassihi, and A. M. Dehnavi, "3d u-net: A voxel-based method in binding site prediction of protein structure," *Journal of Bioinformatics and Computational Biology*, vol. 19, no. 02, p. 2150006, 2021.

[64] V. Iglovikov and A. Shvets, "Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation," Jun 2018, arXiv preprint Computer Vision and Pattern Recognition. [Online]. Available: https://arxiv.org/abs/1801.05746

[65] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[66] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[67] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," 2018.

[68] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," 2019.

[69] M. T. McCann, K. H. Jin, and M. Unser, "Convolutional neural networks for inverse problems in imaging: A review," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 85–95, nov 2017. [Online]. Available: https://doi.org/10.1109%2Fmsp.2017.2739299

[70] A. J. Reader, G. Corda, A. Mehranian, C. da Costa-Luis, S. Ellis, and J. A. Schnabel, "Deep learning for PET image reconstruction," *IEEE Transactions on Radiation and Plasma Medical Sciences*, vol. 5, no. 1, pp. 1–25, 2020.

[71] F. Wang, A. Eljarrat, J. Müller, T. R. Henninen, R. Erni, and C. T. Koch, "Multi-resolution convolutional neural networks for inverse problems," *Scientific Reports*, vol. 10, no. 1, pp. 1–11, 2020.

[72] Z. Zou, T. Shi, Z. Shi, and J. Ye, "Adversarial training for solving inverse problems in image processing," *IEEE Transactions on Image Processing*, vol. 30, pp. 2513–2525, 2021.

[73] B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, and M. S. Rosen, "Image reconstruction by domain-transform manifold learning," *Nature*, vol. 555, no. 7697, pp. 487–492, 2018.

[74] M. P. Heinrich, M. Stille, and T. M. Buzug, "Residual U-net convolutional neural network architecture for low-dose CT denoising," *Current Directions in Biomedical Engineering*, vol. 4, no. 1, pp. 297–300, 2018.

[75] M. P. Reymann *et al.*, "U-net for SPECT image denoising," in *2019 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*. IEEE, 2019, pp. 1–2.

[76] J. Liu, Y. Yang, M. N. Wernick, P. H. Pretorius, and M. A. King, "Deep learning with noise-to-noise training for denoising in SPECT myocardial perfusion imaging," *Medical Physics*, vol. 48, no. 1, pp. 156–168, 2021.

[77] P.-Y. Liu and E. Y. Lam, "Image reconstruction using deep learning," *CoRR*, vol. abs/1809.10410, 2018.

[78] K. Gong, C. Catana, J. Qi, and Q. Li, "Pet image reconstruction using deep image prior," *IEEE transactions on medical imaging*, 2018.

[79] A. Dubbs, J. Guevara, and R. Yuste, "moco: Fast motion correction for calcium imaging," *Frontiers in Neuroinformatics*, vol. 10, p. 6, 2016.

[80] J. K. Min *et al.*, "Rationale and design of the victory (validation of an intracycle CT motion correction algorithm for diagnostic accuracy) trial," *Journal of Cardiovascular Computed Tomography*, vol. 7, no. 3, pp. 200–206, 2013.

[81] "MoCo: SPECT motion correction," https://www.thecardiacsuite.com/moco/, accessed: 2022-09-08.

[82] N. Matsumoto *et al.*, "Quantitative assessment of motion artifacts and validation of a new motion-correction program for myocardial perfusion SPECT," *Journal of Nuclear Medicine*, vol. 42, no. 5, pp. 687–694, 2001.

[83] D. Mitra, D. Eiland, T. Walsh, R. Bouthcko, G. T. Gullberg, and N. Schechtmann, "SinoCor: a clinical tool for sinogram-level patient motion correction in SPECT," in *Medical Imaging 2011: Image Processing*, vol. 7962. SPIE, 2011, pp. 1518–1522.

[84] J. Maier *et al.*, "Deep learning-based coronary artery motion estimation and compensation for short-scan cardiac CT," *Medical Physics*, vol. 48, no. 7, pp. 3559–3571, 2021.

[85] M. A. Al-Masni, S. Lee, J. Yi, S. Kim, S.-M. Gho, Y. H. Choi, and D.-H. Kim, "Stacked U-nets with self-assisted priors towards robust correction of rigid motion artifact in brain MRI," *NeuroImage*, vol. 259, p. 119411, 2022.

[86] W. Jiang *et al.*, "Respiratory motion correction in abdominal MRI using a densely connected U-net with GAN-guided training," Jun 2019, arXiv preprint Image and Video Processing. [Online]. Available: https://arxiv.org/abs/1906.09745

[87] J. Zhang and H. Zuo, "A deep RNN for CT image reconstruction," in *Medical Imaging 2020: Physics of Medical Imaging*, G.-H. Chen and H. Bosmans, Eds., vol. 11312, International Society for Optics and Photonics. SPIE, 2020, p. 113124N. [Online]. Available: https://doi.org/10.1117/12.2549809

[88] L. Fu and B. De Man, "A hierarchical approach to deep learning and its application to tomographic reconstruction," in *15th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, S. Matej and S. D. Metzler, Eds., vol. 11072, International Society for Optics and Photonics. SPIE, 2019, p. 1107202. [Online]. Available: https://doi.org/10.1117/12.2534615

[89] Y. Li, K. Li, C. Zhang, J. Montoya, and G.-H. Chen, "Learning to reconstruct computed tomography images directly from sinogram data under a variety of data acquisition conditions," *IEEE Transactions on Medical Imaging*, vol. 38, no. 10, pp. 2469–2481, 2019.

[90] D. Wu, K. Kim, G. El Fakhri, and Q. Li, "Iterative low-dose CT reconstruction with priors trained by artificial neural network," *IEEE Transactions on Medical Imaging*, vol. 36, no. 12, pp. 2479–2486, 2017.

[91] A. Makhzani and B. J. Frey, "Winner-take-all autoencoders," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper/2015/file/5129a5ddcd0dcd755232baa04c231698-Paper.pdf

[92] H. Lim, I. Y. Chun, Y. K. Dewaraja, and J. A. Fessler, "Improved low-count quantitative pet reconstruction with an iterative neural network," *IEEE Transactions on Medical Imaging*, vol. 39, no. 11, pp. 3512–3522, 2020.

[93] J. Ouyang, K. T. Chen, E. Gong, J. Pauly, and G. Zaharchuk, "Ultra-low-dose pet reconstruction using generative adversarial network with feature matching and task-specific perceptual loss," *Medical Physics*, vol. 46, no. 8, pp. 3555–3564, 2019. [Online]. Available: https://aapm.onlinelibrary.wiley.com/doi/abs/10.1002/mp.13626

[94] I. Häggström, C. R. Schmidtlein, G. Campanella, and T. J. Fuchs, "Deeppet: A deep encoder–decoder network for directly solving the pet image reconstruction inverse problem," *Medical image analysis*, vol. 54, pp. 253–262, 2019.

[95] K. Gong *et al.*, "EMnet: An unrolled deep neural network for PET image reconstruction," in *Medical Imaging 2019: Physics of Medical Imaging*, vol. 10948. International Society for Optics and Photonics, 2019, p. 1094853.

[96] K. Gong, D. Wu, K. Kim, J. Yang, T. Sun, G. El Fakhri, Y. Seo, and Q. Li, "MAPEM-net: an unrolled neural network for fully 3D pet image reconstruction," in *15th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, vol. 11072. International Society for Optics and Photonics, 2019, p. 110720O.

[97] E. Levitan and G. T. Herman, "A maximum a posteriori probability expectation maximization algorithm for image reconstruction in emission tomography," *IEEE Transactions on Medical Imaging*, vol. 6, no. 3, pp. 185–192, 1987.

[98] L. A. Shepp and Y. Vardi, "Maximum likelihood reconstruction for emission tomography," *IEEE Transactions on Medical Imaging*, vol. 1, no. 2, pp. 113–122, 1982.

[99] M. Heideman, D. Johnson, and C. Burrus, "Gauss and the history of the fast fourier transform," *IEEE ASSP Magazine*, vol. 1, no. 4, pp. 14–21, 1984.

[100] G. Strang, "Wavelets," *American Scientist*, vol. 82, no. 3, pp. 250–255, 1994.

[101] R. D. Kent, C. Read, and R. D. Kent, *The acoustic analysis of speech*. Singular Publishing Group San Diego, 1992, vol. 58.

[102] J. Dongarra and F. Sullivan, "Guest editors' introduction: The top 10 algorithms," *Computing in Science & Engineering*, vol. 2, no. 1, pp. 22–23, 2000.

[103] L.-T. Chang, "A method for attenuation correction in radionuclide computed tomography," *IEEE Transactions on Nuclear Science*, vol. 25, no. 1, pp. 638–643, 1978.

[104] E. Biot, E. Crowell, H. Hofte, Y. Maurin, S. Vernhettes, and P. Andrey, "A new filter for spot extraction in n-dimensional biological imaging," in *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*. IEEE, 2008, pp. 975–978.

[105] J. A. Patton and T. G. Turkington, "Spect/ct physical principles and attenuation correction," *Journal of nuclear medicine technology*, vol. 36, no. 1, pp. 1–10, 2008.

[106] W. Sureshbabu and O. Mawlawi, "Pet/ct imaging artifacts," *Journal of nuclear medicine technology*, vol. 33, no. 3, pp. 156–161, 2005.

[107] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[108] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 315–323.

[109] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[110] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[111] G. El Fakhri, A. Kardan, A. Sitek, S. Dorbala, N. Abi-Hatem, Y. Lahoud, A. Fischman, M. Coughlan, T. Yasuda, and M. F. Di Carli, "Reproducibility and accuracy of quantitative myocardial blood flow assessment with 82rb pet: comparison with 13n-ammonia pet," *Journal of Nuclear Medicine*, vol. 50, no. 7, pp. 1062–1071, 2009.

[112] A. Rahmim, O. Rousset, and H. Zaidi, "Strategies for motion tracking and correction in PET," *PET Clinics*, vol. 2, no. 2, pp. 251–266, 2007.

[113] A. Z. Kyme and R. R. Fulton, "Motion estimation and correction in SPECT, PET and CT," *Physics in Medicine & Biology*, 2021.

[114] F. Godenschweger *et al.*, "Motion correction in MRI of the brain," *Physics in Medicine & Biology*, vol. 61, no. 5, p. R32, 2016.

[115] M. F. Callaghan, O. Josephs, M. Herbst, M. Zaitsev, N. Todd, and N. Weiskopf, "An evaluation of prospective motion correction (PMC) for high resolution quantitative MRI," *Frontiers in Neuroscience*, vol. 9, p. 97, 2015.

[116] M. Zaitsev, B. Akin, P. LeVan, and B. R. Knowles, "Prospective motion correction in functional MRI," *NeuroImage*, vol. 154, pp. 33–42, 2017.

[117] M. Andersen, I. M. Björkman-Burtscher, A. Marsman, E. T. Petersen, and V. O. Boer, "Improvement in diagnostic quality of structural and angiographic MRI of the brain using motion correction with interleaved, volumetric navigators," *PLOS One*, vol. 14, no. 5, p. e0217145, 2019.

[118] D. Zerfowski, "Motion artifact compensation in CT," in *Medical Imaging 1998: Image Processing*, vol. 3338. International Society for Optics and Photonics, 1998, pp. 416–424.

[119] R. L. Smith, K. Wells, J. Jones, P. Dasari, C. Lindsay, and M. King, "Toward a framework for high resolution parametric respiratory motion modelling," in *2013 IEEE Nuclear Science Symposium and Medical Imaging Conference (2013 NSS/MIC)*. IEEE, 2013, pp. 1–4.

[120] D. Mitra, D. Eiland, M. Abdallah, R. Bouthcko, G. T. Gullberg, and N. Schechtmann, "SinoCor: motion correction in SPECT," in *Medical Imaging 2012: Image Processing*, vol. 8314. International Society for Optics and Photonics, 2012, p. 831452.

[121] B. Desjardins and E. A. Kazerooni, "ECG-gated cardiac CT," *American Journal of Roentgenology*, vol. 182, no. 4, pp. 993–1010, 2004.

[122] H. Machida *et al.*, "Current and novel imaging techniques in coronary CT," *Radiographics*, vol. 35, no. 4, pp. 991–1010, 2015.

[123] M. S. Nacif, A. Zavodni, N. Kawel, E.-Y. Choi, J. A. Lima, and D. A. Bluemke, "Cardiac magnetic resonance imaging and its electrocardiographs (ECG): tips and tricks," *The International Journal of Cardiovascular Imaging*, vol. 28, no. 6, pp. 1465–1475, 2012.

[124] R. L. Ehman, M. McNamara, M. Pallack, H. Hricak, and C. Higgins, "Magnetic resonance imaging with respiratory gating: techniques and advantages," *American Journal of Roentgenology*, vol. 143, no. 6, pp. 1175–1182, 1984.

[125] S. A. Nehmeh *et al.*, "Effect of respiratory gating on quantifying PET images of lung cancer," *Journal of Nuclear Medicine*, vol. 43, no. 7, pp. 876–881, 2002.

[126] P. Giraud and A. Houle, "Respiratory gating for radiotherapy: main technical aspects and clinical benefits," *International Scholarly Research Notices*, vol. 2013, 2013.

[127] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[128] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9446–9454.

[129] H. Sheikh and A. Bovik, "Image information and visual quality," *IEEE Transactions on Image Processing*, vol. 15, no. 2, pp. 430–444, 2006.

[130] H. R. Sheikh, M. F. Sabir, and A. C. Bovik, "A statistical evaluation of recent full reference image quality assessment algorithms," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3440–3451, 2006.

[131] F. Chollet *et al.* (2015) Keras. [Online]. Available: https://github.com/fchollet/keras

[132] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[133] B. Yang, L. Ying, and J. Tang, "Artificial neural network enhanced bayesian pet image reconstruction," *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, pp. 1297–1309, 2018.

[134] J. Cui, X. Liu, Y. Wang, and H. Liu, "Deep reconstruction model for dynamic pet images," *PloS one*, vol. 12, no. 9, p. e0184667, 2017.

[135] N. Lang, M. Dawood, F. Büther, O. Schober, M. Schäfers, and K. Schäfers, "Organ movement reduction in pet/ct using dual-gated listmode acquisition," *Zeitschrift für Medizinische Physik*, vol. 16, no. 1, pp. 93–100, 2006, schwerpunkt: Bildgebung in der Nuklearmedizin.

[136] A. Gillman, J. Smith, P. Thomas, S. Rose, and N. Dowson, "Pet motion correction in context of integrated pet/mr: Current techniques, limitations, and future projections," *Medical Physics*, vol. 44, no. 12, pp. e430–e445, 2017.

[137] W. Van Elmpt, J. Hamill, J. Jones, D. De Ruysscher, P. Lambin, and M. Öllers, "Optimal gating compared to 3d and 4d pet reconstruction for characterization of lung tumours," *European journal of nuclear medicine and molecular imaging*, vol. 38, no. 5, pp. 843–855, 2011.

[138] L. Livieratos, L. Stegger, P. M. Bloomfield, K. Schafers, D. L. Bailey, and P. G. Camici, "Rigid-body transformation of list-mode projection data for respiratory motion correction in cardiac PET," *Physics in Medicine and Biology*, vol. 50, no. 14, pp. 3313–3322, jul 2005.

[139] X. Zhang, A. Belloso, Y. Yang, M. N. Wernick, P. Hendrik Pretorius, and M. A. King, "A study of deep learning networks for motion compensation in cardiac gated spect images," in *2022 IEEE International Conference on Image Processing (ICIP)*, 2022, pp. 1906–1910.

[140] A. Iyer, C. Lindsay, P. Pretorius, and M. King, "Learning to estimate a surrogate respiratory signal from cardiac motion by signal-to-signal translation," Oct 2021, arXiv preprint Image and Video Processing. [Online]. Available: https://arxiv.org/abs/2208.01034

[141] W. Segars, B. Tsui, E. Frey, G. Johnson, and S. Berr, "Development of a 4-d digital mouse phantom for molecular imaging research," *Molecular imaging and biology : MIB : the official publication of the Academy of Molecular Imaging*, vol. 6, pp. 149–59, 05 2004.

[142] W. P. Segars and B. M. W. Tsui, "Mcat to xcat: The evolution of 4-d computerized phantoms for imaging research," *Proceedings of the IEEE*, vol. 97, no. 12, pp. 1954–1968, 2009.

[143] Z. Yang, B. A. French, W. D. Gilson, A. J. Ross, J. N. Oshinski, and S. S. Berr, "Cine magnetic resonance imaging of myocardial ischemia and reperfusion in mice," *Circulation*, vol. 103, no. 15, pp. e84–e84, 2001.

[144] W. Segars, D. Lalush, and B. Tsui, "Modeling respiratory mechanics in the mcat and spline-based mcat phantoms," *IEEE Transactions on Nuclear Science*, vol. 48, no. 1, pp. 89–97, 2001.

[145] W. Segars, B. Tsui, D. Lalush, E. Frey, M. King, and D. Manocha, "Development and application of the new dynamic nurbs-based cardiac-torso (ncat) phantom." *JOURNAL OF NUCLEAR MEDICINE*, vol. 42, p. 23, 05 2001.

[146] W. P. Segars, S. Mendonca, G. Sturgeon, and B. M. W. Tsui, "Enhanced 4d heart model based on high resolution dual source gated cardiac ct images," in *2007 IEEE Nuclear Science Symposium Conference Record*, vol. 4, 2007, pp. 2617–2620.

[147] E. Avelar *et al.*, "Left ventricular hypertrophy in severe obesity: interactions among blood pressure, nocturnal hypoxemia, and body mass," *Hypertension*, vol. 49, no. 1, pp. 34–39, 2007.

[148] N. Aghakhan Olia, A. Kamali-Asl, S. Hariri Tabrizi, P. Geramifar, P. Sheikhzadeh, S. Farzanefar, H. Arabi, and H. Zaidi, "Deep learning–based denoising of low-dose spect myocardial perfusion images: quantitative assessment and clinical performance," *European journal of nuclear medicine and molecular imaging*, pp. 1–15, 2022.

[149] A. J. Ramon, Y. Yang, P. H. Pretorius, K. L. Johnson, M. A. King, and M. N. Wernick, "Improving diagnostic accuracy in low-dose spect myocardial perfusion imaging with convolutional denoising networks," *IEEE transactions on medical imaging*, vol. 39, no. 9, pp. 2893–2903, 2020.

[150] X. Wang, L. Zhou, Y. Wang, H. Jiang, and H. Ye, "Improved low-dose positron emission tomography image reconstruction using deep learned prior," *Physics in Medicine & Biology*, vol. 66, no. 11, p. 115001, 2021.

[151] A. Ben-Cohen, E. Klang, S. P. Raskin, S. Soffer, S. Ben-Haim, E. Konen, M. M. Amitai, and H. Greenspan, "Cross-modality synthesis from ct to pet using fcn and gan networks for improved automated lesion detection," *Engineering Applications of Artificial Intelligence*, vol. 78, pp. 186–194, 2019.

[152] C.-B. Jin, H. Kim, M. Liu, W. Jung, S. Joo, E. Park, Y. S. Ahn, I. H. Han, J. I. Lee, and X. Cui, "Deep ct to mr synthesis using paired and unpaired data," *Sensors*, vol. 19, no. 10, p. 2361, 2019.

[153] V. Kearney, B. P. Ziemer, A. Perry, T. Wang, J. W. Chan, L. Ma, O. Morin, S. S. Yom, and T. D. Solberg, "Attention-aware discrimination for mr-to-ct image translation using cycle-consistent generative adversarial networks," *Radiology: Artificial Intelligence*, vol. 2, no. 2, p. e190027, 2020.

Appendix A

Publications

Journal

- Pan H, **Chang H**, Mitra D, Gullberg GT, and Seo Y, "Sparse domain approaches in dynamic SPECT imaging with high-performance computing," Am J Nucl Med Mol Imaging 20;7(6):283-294, 2017 PMID: 29348983; PMCID: PMC5768923.

- Mitra D, Abdalah M, Boutchko R, **Chang H**, Shrestha U, Botvinick E, Seo Y, and Gullberg GT, "Comparison of sparse domain approaches for 4D SPECT dynamic image reconstruction," Medical Physics, 45(10):4493-4509, 2018. doi: 10.1002/mp.13099. Epub 2018 Aug 31. PMID: 30027577; PMCID: PMC6211286.

Conf Proceedings

- **H. Chang**, D. Mitra, U. Shrestha, G. T. Gullberg and Y. Seo, "Parameters Estimation Directly from Sinograms with Neural Networks," IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), Manchester, UK, 2019, pp. 1-5, doi: 10.1109/NSS/MIC42101.2019.9059984.

- **H. Chang**, R. Smith, S. Paisey, R. Boutchko and D. Mitra, "Deep Learning Image Transformation under Radon Transform," IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), Boston, MA, USA, 2020, pp. 1-3, doi: 10.1109/NSS/MIC42677.2020.9507793.

- **H. Chang**, and D. Mitra, "Motion estimation and motion-corrected reconstruction with inverse Radon transformation using deep learning," Mid-Winter Meeting of the Soc. Of Nucl. Med. And Mol. Imag., Tampa, FL, US, 2020.

- **H. Chang**, V. Kobzarenko, R. Smith, S. Paisey, and D. Mitra, "Affine Motion Correction Using Deep Learning," IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), Yokohama, Japan, 2021.

- W. Stern, **H. Chang**, R. Boutchko, U. Shrestha, Grant T. Gullberg, Y. Seo, and D. Mitra, "Use of Conventional Late Imaging Protocol for Dynamic SPECT Imaging," Annual Conference of Soc. Of Nucl. Med. In Med. Imag, Chicago, IL, US, 2023.

Under communication

- **H. Chang**, V. Kobzarenko, and D. Mitra, "Inverse Radon Transform with Deep Learning: an application in cardiac motion correction."

- **H. Chang**, R. Smith, S. Paisey, and D. Mitra, "Efficient Motion-corrected Cardiac PET Image Reconstruction by Deep Learning without Real Training Data."