Towards safer X-rays

Towards Safer X-rays

By Paria Kargar Samani,

A Thesis Submitted to the School of Graduate Studies in the Partial Fulfillment of the Requirements for the Degree Masters of Applied science in Electrical Engineering

McMaster University

Masters of Applied science in Electrical Engineering (2019)

Hamilton, Ontario (Department of Electrical and Computer Engineering)

TITLE: Towards Safer X-rays

AUTHOR: Paria Kargar Samani (McMaster University)

SUPERVISOR: Dr. Shahram Shirani

NUMBER OF PAGES: xvii, 79

Abstract

High-intensity X-ray radiation through the human body can cause damage to cells, increasing the chance of complications such as cancer. A possible solution is lowering the dosage of radiation. In low dose CT (LDCT) images, fundamental structures are still easily identifiable. However, noise and other additional artifacts are introduced. Removing the visual effects of artifacts caused by lowering radiation dose has been an active area of research in the last few years. Recently, deep learning approaches have demonstrated impressive performance for LDCT denoising. In this thesis, we propose a new machine learning-based approach for LDCT noise reduction that outperforms other methods.

Deep Learning is based on the idea of stacking many layers of neurons together. Over the past years, deep learning researchers have successfully optimized the performance of neural networks by stacking more layers. With the growing availability of high-performance GPUs as well as more massive datasets, deep learning technology has proven very useful. However, deeper networks are more challenging to train, not because of their computational cost, but due to the difficulty of propagating gradients through so many layers.

Deeper neural networks are more complex to train. We present a residual framework to ease the training of networks that are substantially deeper than the others. Residual learning means each subsequent layer in a deep neural network is only responsible for, fine-tuning the output from a previous layer, which is possible only by adding a learned "residual" to the input. This method differs from a more traditional approach where each layer had to generate the total desired output. By using residual learning in LDCT denoising, we prevent degradation of training accuracy once traversing the network and, increase the training pace.

Another aspect of our work is using Generative Adversarial Network (GAN), which is a framework for estimating generative models via an adversarial process. We simultaneously train two models a generative model G, that we are using for generating Normal Dose CT (NDCT) and a discriminative model D that estimates the probability that a sample came from the training data rather than G. GANs' potential is enormous since they can learn to mimic any distribution of data.

The novelty of our approach is in combining Residual learning and GAN. For training a convolutional neural network (CNN), a large amount of data is needed. We address this problem by using patch coding. Inspired by the idea of deep learning, we combine the autoencoder, deconvolutional network, and skip connections into residual learning. One motivation for skipping over the layers is to avoid the problem of gradient vanishing. Our experiments show that our method outperforms recent works on LDCT image denoising in terms of Peak SNR (PSNR), and SSIM.

Acknowledgements

I would like to express my special thanks to my supervisor, Dr. Shahram Shirani of the Department of Electrical and Computer Engineering at McMaster University. Dr. Shirani was always available whenever I needed assistance about my research and projects. The completion of my adventure at McMaster University could not have been possible without Dr. Shirani's guidance and helpful suggestions. I would also like to express my deep appreciation to all of my lab mates from Multimedia Signal Processing Laboratory at McMaster University, especially Yasamin Fazliani who helped a lot by reviewing this thesis.

Finally, I would like to thank my family and friends Dr. Anis Fard Mousavi and Dr. Amir Kargar Samani, Ilya Kargar Samani, Negar Arabzadeh, and Parisa Mahvelati Shams Abadi, for providing support throughout my journey at McMaster. This couldn't have been possible without you all and I dedicate this thesis to you.

Contents

List of Figures

List of Tables

List of Abbreviations

1-D	**1 D**imention
2-D	**2 D**imention
3-D	**3 D**imention
AWGN	**A**dditive **W**hite **G**aussian **N**oise
CCD	**C**harge **C**oupled **D**evice
CM3D	**B**lock **M**atching **3 D**imentional filtering
CMOS	**C**omplementary **M**etal **O**xide **S**emiconductor
CNN	**C**onvolutional **N**eural **N**etwork
CT	**C**omputed **T**omography
DCT	**D**iscrete **C**osine **T**ransform
DeCNN	**D**enoising **C**onvolutional **N**eural **N**etwork
FBP	**F**iltered **B**ack **P**rojection
FFT	**F**ast **F**ourier **T**ransform
GAN	**G**enerative **A**dversarial **N**etwork
HR	**H**igh **R**esolution
ID	**ID**entification
ILSVRC	**I**mageNet **L**arge **S**cale **V**isual **R**ecognition **C**hallenge
IR	**I**mage **R**estoration
JPEG	**J**oint **P**hotographic **E**xperts **G**roup
JS	**J**enson-**S**hannon
KL	**K**ullback-**L**eibler

LDCT	**L**ow **D**ose **C**omputed **T**omography
LR	**L**ow **R**esolusion
LReLU	**L**eaky **Re**ctified **L**inear **U**nit
LSSC	**L**earned **S**imultaneous **S**pare **C**oding
ML	**M**achine **L**earning
MLE	**M**aximum **L**ikehood **E**stimation
MRI	**M**agnetic **R**esonance **I**maging
MSE	**M**ean **S**quared **E**rror
NCSR	**N**onlocally **C**enteralized **S**pare **R**epresentation
NDCT	**N**ormal **D**ose **C**omputed **T**omography
NLGC	**N**on-**L**inear **G**aussian filtering **C**hain
NN	**N**eural **N**etwork
PSNR	**P**eak to **S**ignal **R**atio
RED-CNN	**R**esidual **E**ncoder **D**ecoder **C**onvolutional **N**eural **N**etwork
ReLU	**Re**ctified **L**inear **U**nit
RES-GAN	**Res**idual Learning **G**enerative **A**dversarial **N**etwork
SISR	**S**ingle **I**mage **S**uper **R**esolusion
SR	**S**parse **R**epresentation
SSIM	**S**tructural **S**imilarity **I**ndex **M**ethod
Tanh	Hyperbolic Tangent
TV	**T**otal **V**ariation
WGAN	**W**asserstein **G**enerative **A**dversarial **N**etwork
WNNM	**W**eighted **N**uclear **N**orm **M**inimization

List of Symbols

b_l	bias of the l_{th} layer
$b_l{}'$	bias of the deconvolutional l_{th} layer
c	Constant derivative
c	contrast comparison function
D_l	output of the lth layer
$D(x;\theta_d)$	mapping of discriminator to data space
e_g	generator error
E	error
f	reference image
F_l	output feature map
g	test image
$G(z;\theta_g)$	mapping of generator to data space
h	binary cross-entropy
H	Degradiation matrix
i	Horizontal position of the element in 2-D space
I	Intensity
I	Network input
I_0	Intensity at zero distance
I_{LD}	low-dose image patch
I_{ND}	corresponding normal dose computed tomography image patch
j	Vertical position of the element in 2-D space

$JS()$	Jenson-Shannon divergence
k	number of validations
K	2-D kernal/filter
l	luminance comparison function
L	Interval length
M	Total absorption
MAX_I	maximum possible pixel value of an image
N	noise
p_g	probability distribution by generator
$p_z(Z)$	input noise variable
r	Distance
s	structure comparison function
W	Weight matrix
EM	Earth Mover/Wasserestein distance
W_l	convolutional filter of the l_{th} layer
$W_l{}'$	weight of deconvolutional l_{th} layer
x	Distance
x	Clean image
$\widehat{x}$	Estimation of the clean image
X	Input matrix
X	NDCT image matrix
y	Corrupted image
Y	Output matrix
Y	LDCT image matrix
z	Noise of standard deviation
z	simple distribution such as uniform or Gaussian distribution
Z	noise matrix

θ	angle
θ	distribution parameter
λ	Trade-off parameter
μ	Absorption coefficient
μ	mean luminance
σ	Standard deviation

Declaration of Authorship

I, Paria Kargar Samani, declare that this thesis titled, "Towards Safer X-rays" and the work presented in it are my own. I confirm that:

- Chapter Introduction
- Chapter Literature review
- Chapter Residual Learning Approach
- Chapter GAN approach
- Chapter RES-GAN approach
- Chapter Conclusion and Future work
- Writing the thesis and implementing the codes

were done wholly by me while I was in candidature for a Masters degree at this University.

Chapter 1

Introduction

1.1 Introduction

The need for an X-ray imaging arises for diagnosing a bone fracture or occasionally other forms of complication in the body. Similar to radio waves, X-rays are a form of electromagnetic radiation but at a higher frequency. This translates into X-rays conveying with higher energy enabling them to pass through human soft tissues and internal organs due to their inability to absorb high energy radiation. On the other hand, denser tissues such as bones absorbing the radiation. The difference in absorption is utilized to form a picture of the imaged area. Although X-rays remain one of the most traditional diagnostic imaging techniques, the high beam radiation through the human body can cause damage to cells, increasing the chance of complications such as cancer. Brenner et al. [3] provides a thorough study on the ascending number of CT scans, dose amount, and the increase in the risk of cancer associated with X-ray type imaging.

Additionally, X-ray radiation is employed for generating Computed Tomography (CT) scan, in which a series of X-ray images are taken from multiple angles of the tissue of interest. CT scans provide a more detailed capture making them a suitable choice for diagnosing a wider range of internal injuries. However, this also implies the higher risks

involved with CT scan compared to plain X-ray due to longer exposure time to ionizing radiation. According to [3], the risk involved with CT scans depends on various factors including tube current and scanning time.

In recent years, there has been an increasing use of CT; according to [3], an average of over 62 million CT scans are taken in the United States per year which is about 20 times more than what it was in 1980 and at least 4 million of this number was CT scans taken from minor patients.

One of the approaches proposed to reduce the risk of CT imaging is lowering the dosage of radiation. In low dose CT (LDCT) imaging, fundamental structures are still easily identifiable. However, noise and additional artifacts are introduced to CT images [40] as a drawback. Several studies have been conducted to improve the quality of LDCT images. One common technique in the literature is to denoise LDCT images. This thesis aims at developing a deep neural network solution for denoising LDCT scans. The following is how the rest of this chapter proceeds: an introduction to CT scan is given, followed by elaborating on the risks involved with X-ray imaging; concluding the chapter by description of challenges in denoising LDCT images and proposing a solution for tackling them by deploying Convolutional Neural Network(CNN). Sections 1.4 and 1.5 include a brief description of the Neural Networks (NN), and more specifically CNN.

1.2 How CT-scan works

CT scan is a procedure leveraging X-ray radiation to create cross-sectional images that are more thorough than conventional X-ray images. While conventional X-ray generators use a fixed tube sending X-ray in only one direction, CT scanners use motorized X-ray source that shoots narrow beams of X-ray as it rotates around the body. There are digital X-ray detectors, located directly on opposite side of the X-ray source[42]. The

patient lies down on the metal (aluminum or silver) tray and gets pushed through a hoop in a wide annular apparatus, as shown in Figure 1.1. As an X-ray emitting source rotates around the annulus as well as the patient's body, the detectors capture the X-ray on the opposite side of where it is projected, and transmit it to a computer for processing purposes.

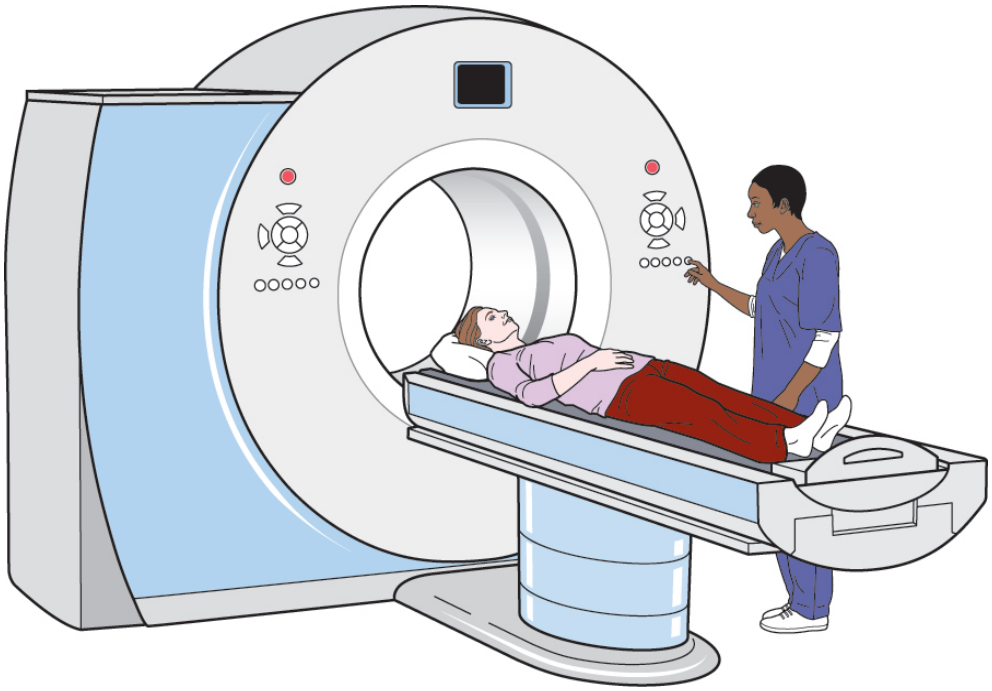


FIGURE 1.1: CT imaging
Image by: https://www.macmillan.org.uk/

1.2.1 Processing a CT scan

Each time the X-ray source completes one full rotation, the CT computer uses sophisticated mathematical techniques to construct a 2D image slice of the patient's organ. Image slices can either be displayed individually or stacked together to generate a 3D reconstruction of the organ. The algebraic procedure to describe what a slice consists of is as follows:

Figure 1.2 presents a simplified version of a body, each of its nine squares representing a particular organ type. As shown in this figure, only a fraction of emitted X-ray radiation travels through and is detected. In Figure 1.2, taking the first square on the top left only 1/1000 of X-ray units are passed, representing a bone structure. Next square represents a fluid type texture allowing 1/10. Finally, the last square relates to a muscle tissue, passing 1/100 of units through.

Taking each of the top squares in figure 1.2 as A, B, and C in the described order, we can define the dampening factor for each tissue type and identify the total X-ray attenuation via the following equation:

$$A \times B \times C = \frac{1}{1000,000} \tag{1.1}$$

In order to find the exact dampening amount in the square, we have to project more X-ray in every direction, including horizontally, vertically, and diagonally. Dampening of X-rays can be expressed as a "line integral"; the sum of the absorbed amount along the path of the beam. The challenge is to determine the patterns of tissue X-ray absorption from the detected X-ray[13].

Finding the texture of each organ type (each square in the example given by figure 1.2) is achievable by number of equations. However, in ones' body, millions of points have to be identified, and the algebraic procedure tends to be computationally expensive [14]. The current method of solving such equation, uses Radon's method for finding dampening amount $I(x)$ at point x, as illustrated in figure 1.3. In mathematical terms:

$$I(x) = I_0 exp(-\int \mu dx) \tag{1.2}$$

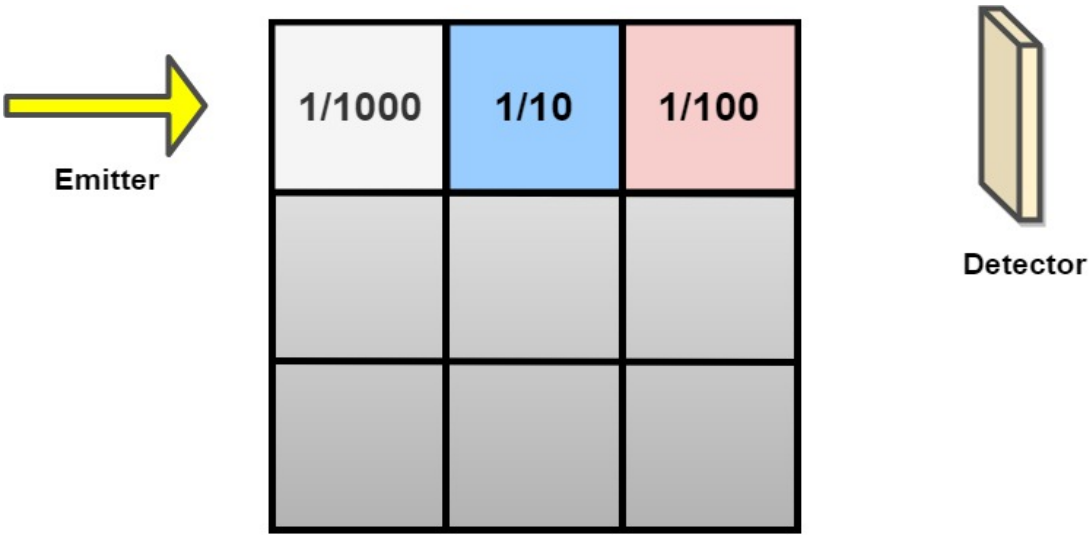


FIGURE 1.2: An example to explain how the X-ray imaging works, gray square represents bone texture, on its right blue square represents a fluid structure and finally red square displays a muscle

In this equation, 1.2 I_0 is intensity at $x = 0$ and μ is the measure of absorption. Therefore, Equation 1.3 demonstrates the rate of attenuation with respect to distance:

$$\frac{dI}{dx} = -\mu I \tag{1.3}$$

So far, we assumed that the absorption coefficient μ is constant. We can generalize the equation by varying the absorption coefficient as a function of distance x; i.e. for a non-homogeneous body. Yielding into a more general case presented in Equation 1.4:

$$I(x) = I_0 exp(-\int_0^x \mu(x)dx) \tag{1.4}$$

If we consider the total absorption M by a material bounded to an interval $[a, b]$ with $0 \leq a < b \leq L$, then:

$$M = \int_a^b \mu dx \tag{1.5}$$

Expanding this into two dimensions, a slightly different approach is taken. Given the total amount of absorption for every cross-section through the body, we can construct the absorption coefficient $\mu(x, y)$ as a function of the position. Johann Radon was the first who showed this in 1917. To reconstruct the structure completely, using Figure 1.2 we need X-rays along every line through the body, which is hard to achieve. However, we can come close to an exact picture by having a broad set of lines.

Any line $\mathcal{L}$ in the xy-plane can be specified by its perpendicular distance r from the origin and the angle θ of the perpendicular. Then any point a on the line is given by Equation 1.6 as follows with s varying along the line.

$$a(x, y) = (r\cos\theta - s\sin\theta, r\sin\theta + s\cos\theta) \tag{1.6}$$

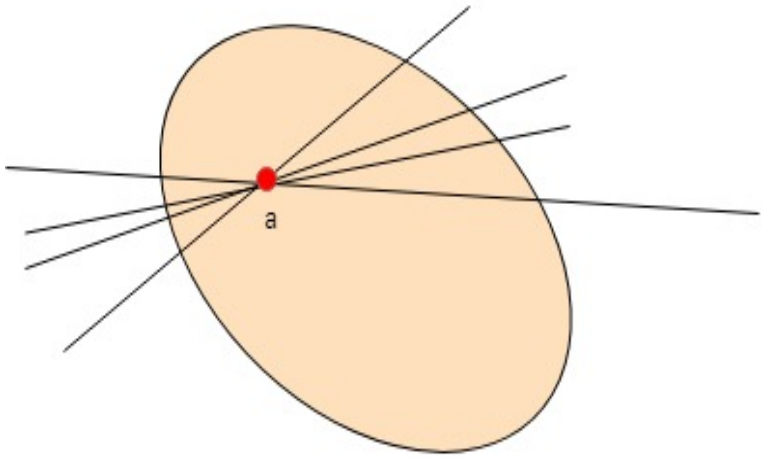


FIGURE 1.3: Illustration of Radon effect, a is an example point that we are trying to measure its absorption and find its texture according to that

If we consider the absorption of an X-ray beam along a line L, we must integrate with respect to s. The result will depend on r and θ:

$$M(r,\theta) = \int \mu(r\cos\theta - s\sin\theta, r\sin\theta + s\cos\theta)ds \tag{1.7}$$

The Radon transform is a function of the polar coordinates (r,θ). It is a linear operation with respect to the function $\mu(x,y)$ being transformed. A graph of $M(r,\theta)$ with r and θ on orthogonal cartesian axes is called a projection data or sinogram; which depicts all the data from a 2D CT scan.

1.2.2 Risk Definition

Each X-ray procedure has a different level of risk associate depending on the type and amount of radiation as well as the body part being exposed. Generally, CT is identified as the most hazardous procedure among X-rays and more specifically abdominal CT, puts the body in the most exposed state of radiation. According to statistics provided by [3]and [17], the two most common organs requiring CT scans are head and abdomen. Consequently, we are focusing on abdominal CT.

A major risk involved with general X-ray radiation is altering healthy body cells into cancerous ones, while inducing immediate side effects such as vomiting, bleeding, fainting, hair and skin loss.

A measure is needed to quantify the risk of exposure. Traditionally, the amount of radiation one receives is measured in Sieverts (Sv), or Grays. One Gray is 1 joule of radiation energy absorbed per kilogram; however the effective dose measures in Sieverts (for X-ray radiation 1 mSv = 1 mGy)[38]. According to [3], every single person receives an average 2 mSv from the background per year.

It is worth mentioning that although organ dose is the ultimate desired quantity for risk estimation, CT dose remains the closest measurable quantity mapping the X-ray exposure risk.

1.3 Problem Description

A possible approach for decreasing the risks involved with CT is lowering the dosage of radiation by reducing the X-ray tube current or shortening the exposure time[38]. It is believed that exposure to lower dosage of radiation reduces the risk of health complications. In low dose CT (LDCT) images, organ structures are still easily identifiable. However, noise and other additional artifacts are introduced to CT images [40] as a drawback. A substantial amount of undergoing research is carried out in the field of image denoising, and more specifically denoising medical images have drawn a lot of attention recently. Some research works have already been implemented in this area to solve the problem and reproduce images as close as possible to normal dose CT (NDCT) images from LDCT.

Image Restoration (IR) is a fundamental problem in the field of image processing involving the recovery of clean image x from its corrupted perception y as stated by the following relation:

$$y = Hx + z \tag{1.8}$$

[53] where H is the degradation matrix, z is additive noise of standard deviation σ [53]. Depending on the matrix H the image restoration problem differs.

In the scope of this thesis, our image restoration challenge is of a denoising nature; hence H is an identity matrix and the objective is to obtain an estimate of x denoted by $\hat{x}$ as the following:

$$\hat{x} = \underset{x}{argmax} \quad logP(y|x) + logP(x) \tag{1.9}$$

where $logP(y|x)$ is log-likelihood of observation y, $logP(x)$ gives us priori of x and it is independent of y. Equation 1.9 can be expanded into the following equation:

$$\hat{x} = \underset{x}{argmin} \quad \frac{1}{2}\|y - Hx\|^2 + \lambda\phi(x) \tag{1.10}$$

The above equation contains two terms, fidelity term, $\frac{1}{2}\|y - Hx\|^2$, and regularization term (or priori term), $\phi(x)$. Additionally, λ is a trade-off parameter. There are two approached for solving Equation 1.10:

- Model-based optimization
- Learning methods

The main difference between model-based optimization method and discriminative learning method is that, the former is flexible to handle various IR problems and solves for the Equation 1.10, with computationally complex algorithms. However, the latter method provides a solution in order to learn the function ϕ and requires training datasets with certain degradation matrices to achieve that, making them limited to specialized tasks. However, with new variable splitting techniques it is possible to handle fidelity and regularization terms separately. More specifically, the latter is the only factor corresponding to our denoising subproblem[53]. The ultimate objective is to minimize the cost function $l(\hat{x}, x)$ as denoted in Equation 1.11[53]:

$$\underset{\phi}{min} \quad l(\hat{x}, x) \quad s.t \quad \hat{x} = \underset{x}{argmin} \quad \frac{1}{2}\|y - Hx\|^2 + \lambda\phi(x;\theta) \tag{1.11}$$

Therefore, we train a Convolutional Neural Network(CNN) as a denoiser and integrate it into our model-based optimization method. To fully understand how a CNN works, we will discuss the fundamentals of general Neural Networks (NN), in upcoming subsection1.4 and followed by an introduction to CNN.

1.4 Introduction to Neural Networks

The word Neural Network (NN) is inspired by the term *neuron*, which refer to neural processing units that are interconnected with one another. By leveraging weights and biases, a neural network could mimic how the biological brain's neurons work. In a basic NN, neurons are arranged in three main layers including: input layer, hidden layer, and output layer. Figure 1.4 shows a simplified model of a NN with three layers of input, hidden, and output[23]. In this figure, X can be a matrix of multiple inputs, and Y can be a matrix of multiple or single outputs. For simplicity purposes, only three weights are shown in the image.

The hidden layers, which may contain one or multiple layers perform the complex calculations. The procedures taking place within these layers may not be visible to end users; unlike the input being fed in to the network and the network's output as the end result. Each layer consists of one or a number of nodes, which the computation takes place in the following manner: a node linearly combines input with a set of coefficients, the weights that either amplify or damper the input to that node by being multiplied to and add a bias. A node receiving inputs from two or more nodes can determine which input requires more attention and assign weights accordingly[23]. These input-weight products are summed, then passed through a node's so-called activation function. If the signals pass through, the neuron has been "activated" as illustrated in figure 1.5.

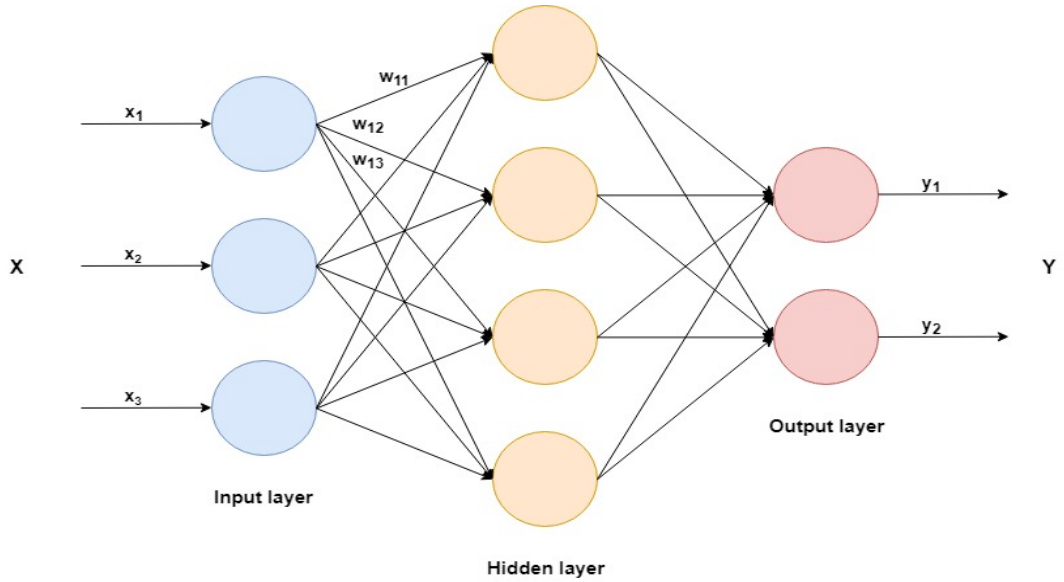


FIGURE 1.4: A simplify model of a neural network, consist of an input, a hidden and an output layer. X shows the inputs and Y shows the outputs also w is for weights

To commence the training procedure, the weights are usually initialized randomly. The training carries on to the point where the parameters are capable of producing accurate classification or prediction. A trained model is the collection of the learned weights aiming to model a relationship that can produce a data structure [31].

The error is defined as the difference between the ground truth and what is estimated through the model. The weights are adjusted with the objective of lowering the error.

The sum of the nodes get passed through a non-linear activation function e.g. sigmoid, tanh, or ReLU in order to prevent increasing without a limit. We are discussing the nature and purpose of activation functions further in section 1.5.2.

For adjusting the weights according to the error we use an optimization function, *gradient descent* being one of the commonly used algorithms. The relationship between network Error and each of those weights is a derivative, $\frac{dE}{dw}$, that measures the degree to which a slight change in weight causes a slight change in the error.

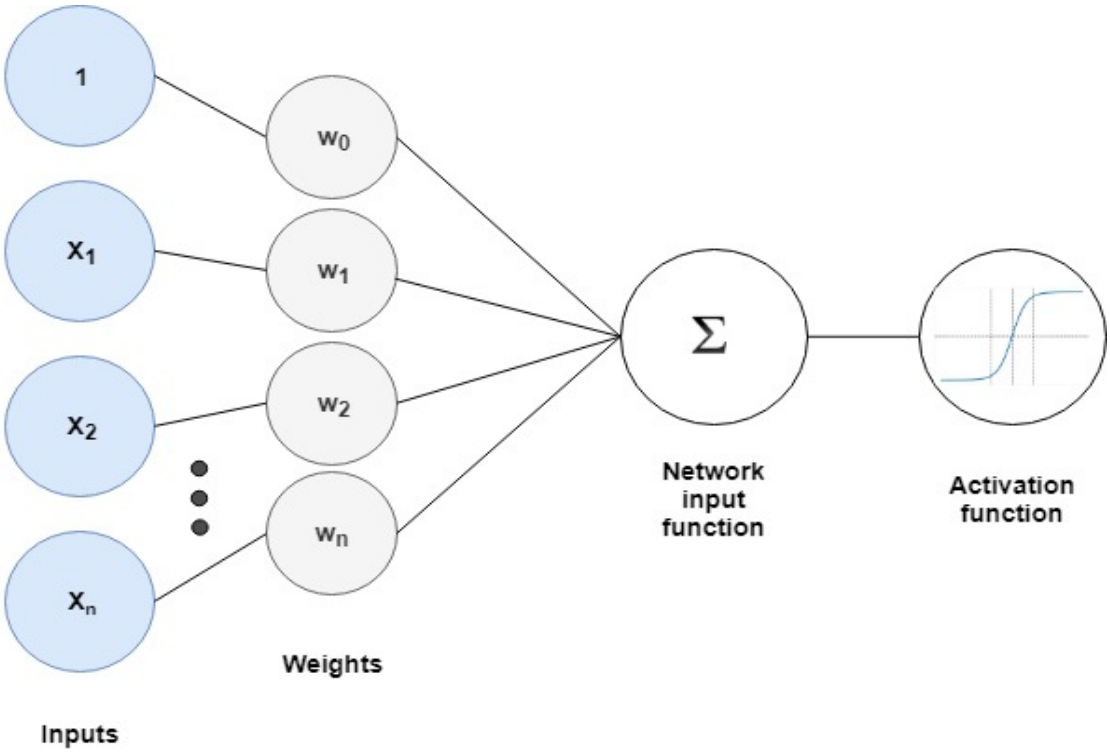


FIGURE 1.5: A hidden layer structure, including the input nodes, corresponding weights and summation node following an activation function

1.5 Convolutional Neural Network

Convolutional Neural Networks (CNN) constitute a class of neural networks that expand convolution in general matrix multiplication in at least one of their layers [18]. There are three main part in a typical CNN layer:

- Performing several parallel convolutions
- Going through a non-linear activation called detector stage
- Applying a pooling function affecting the output layer[18]

The following subsections, a brief discussion of each procedure is given.

1.5.1 Convolution

In CNN, the first argument to the convolution operation is called input and the second one is a kernel or filter. The output can be referred to as a feature map. A multidimensional array of data along with the kernel is often called a tensor. The following equation is called cross-correlation function for a two-dimension convolution:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n) K(m,n) \tag{1.12}$$

In this equation, I is the input to the network, K is a 2-D kernel/filter, i and j are the horizonal and vertical positions of the element in 2-D space. To better understand the equation, Figure 1.6 demonstrates the convolutional procedure. In this example, the filter size is 2×2, and our step size/pace is considered as 1, which means after the operation takes place on the first window, the second one only slides by one step [18]. Each convolutional layer includes several numbers of filters that are defined by a specific dimension and convolves the input image. Therefore an output is given when a given input is convolved with a particular filter, i.e. the output is a matrix of pixels consisted of the input pixels' convolution.

1.5.2 Activation functions

In artificial NN, activation functions play an important role in converting an input of a neuron to an output for being passed to the next neuron. The output value of a neuron can be anything ranging $-\infty$ to $+\infty$. For a neuron to determine to either fire or dismiss a calculated output is to leverage an activation function.

Depending on the application, three different classes of activation function exist: step, linear, and nonlinear. The most basic function of all is known to be the step function,

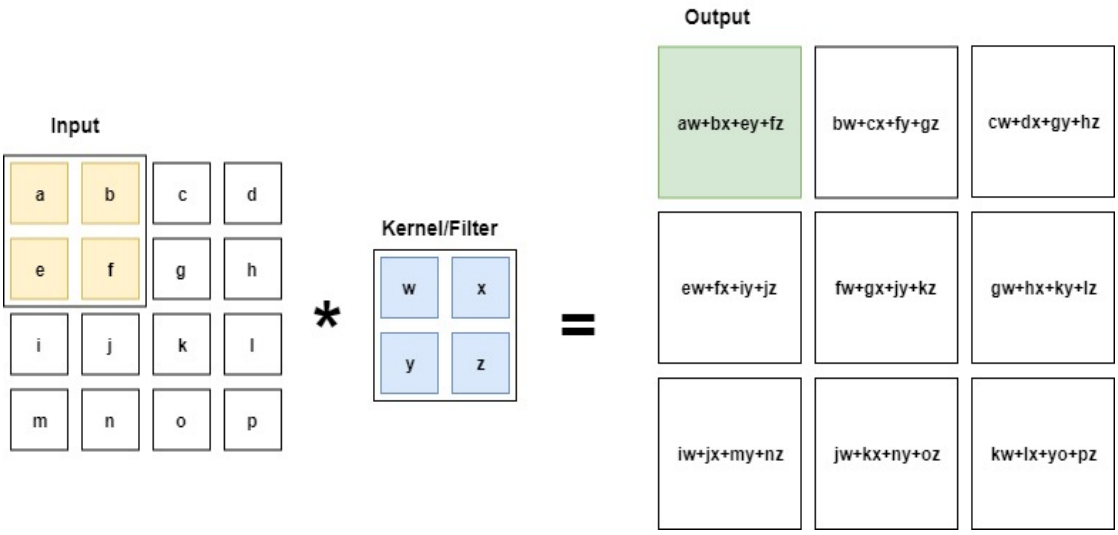


FIGURE 1.6: Yellow window shows the first input matrix that is convolved with the blue filter and the output of convolution is the green window

which is a threshold-based function. If the output value is above a specific threshold, the neuron is declared as *activated*; otherwise, it is not.

Where an activation proportional to the input is desired a linear activation function is applied. This provides a range of activation outputs, as opposed to a binary activation in a step type activation function. The above-mentioned function can be mapped as a linear function brought below:

$$A = cx \tag{1.13}$$

with input x, and a constant derivative c; i.e. the gradient is independent of x and it is a constant descent. If there is an error in prediction, the adjustments made through backpropagation are constant and not depending on the changes in input. Backpropagation is a procedure following a gradient descent approach that exploits the chain rule. Also, no matter how many layers we have, if all are linear, the final activation function of the last layer is nothing but just a linear function of the input of the first layer.

The third class of activation functions, are the non-linear functions. In case of a

significantly complex mapping of inputs to output/s, non-linear activation functions may be deployed. Hence by using a nonlinear activation, we can generate nonlinear mappings from inputs to outputs, yet assuring its differentiability.

Consequently, we can use backpropagation optimization strategy, which includes propagating backwards in the network to compute gradients of error (or loss) concerning the given weights. We then accordingly optimize weights using gradient descend or any other optimization technique to reduce error.

In the following sections, there is a brief overview of some of the widely used activation functions in the scope of CNN.

Sigmoid

The most commonly used activation function is the sigmoid function. The equation below shows how a sigmoid function is defined:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{1.14}$$

As can be depicted from the above equation, sigmoid function gives a 0 as x approaches $-\infty$ and outputs a 1 as it approaches $+\infty$. However, there exists a major disadvantage involved with sigmoid activation functions called the vanishing gradient problem. When a neuron activation saturates close to 0 or 1, the gradient at the regions is very close to 0 during backpropagation. The local gradient affects the entire procedure; so if the local gradient is minimal, it will slowly vanish the gradient and almost a zero signal will flow through the neuron. The other disadvantage of sigmoid function is that the output is not zero centered, i.e. if the output value of the function is positive, the gradient of the weights come out as either all positive or all negative; hence forcing the gradient updates

fluctuate more aggressively in either directions resulting in a more complex optimization process.

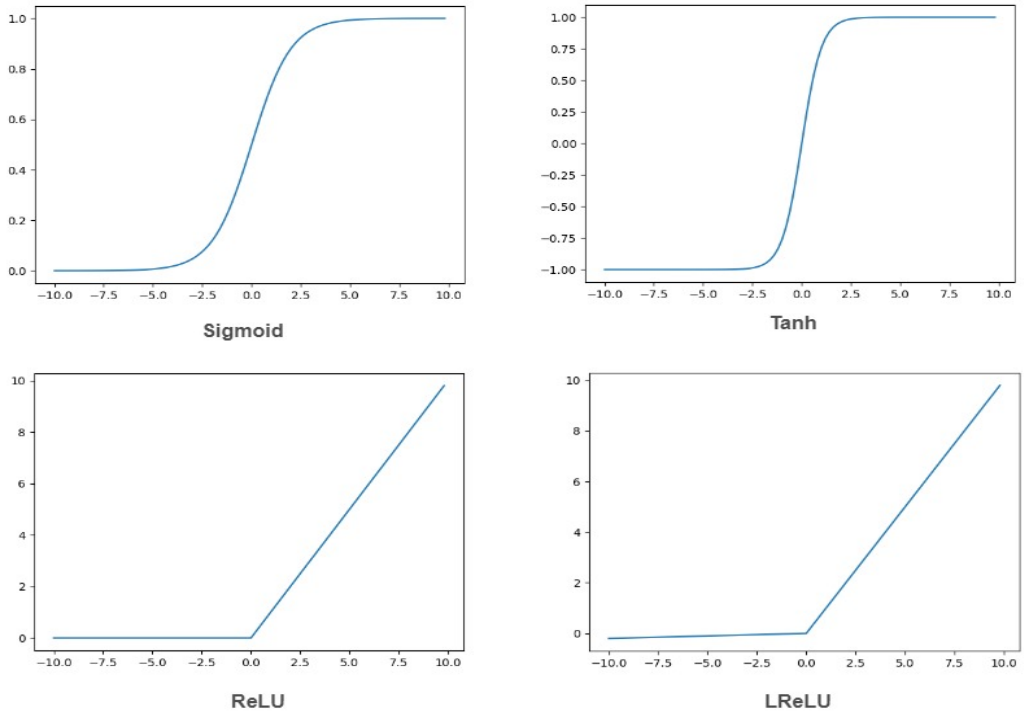


FIGURE 1.7: The most commonly used activation functions for CNN

Tanh

The Hyperbolic tangent function ($Tanh$) is another common activation function, defines as the following[18]:

$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{1.15}$$

This function is zero-centered, therefore creates a smoother optimization process in comparison with sigmoid function. However, similar to sigmoid, hyperbolic tangent function also suffers from the vanishing gradient problem. [30]

ReLU

ReLU (Rectified Linear Unit) activation function became a popular choice in deep learning. ReLU was mainly deployed to solve the vanishing gradient problem mentioned above [1]. The following describes the functionality of ReLU:

$$f(x) = \begin{cases} 0 \quad for \quad x < 0 \\ x \quad for \quad x \geq 0 \end{cases} \tag{1.16}$$

As long as input value is greater than zero, regardless of how large it is, the gradient of the activation function will be 1. This solves the vanishing gradient problem present in the sigmoid activation function.

In the case of classification, the ReLU is only used in hidden layers, and the output layer usually applies a softmax function. Softmax provides a probability for different classes, and a linear function for regression since the signal goes through unchanged. A disadvantage to ReLU is that during the training, some units would be fragile and die. It means a significant gradient flowing through a ReLU neuron could cause a weight update that blocks the activation on any other data point. Therefore the gradient flowing through a neuron will always be equal to zero from that point onwards. [30]

LReLU

Leaky ReLU is a modification of ReLU which replaces the zero part of the domain in $[-\infty, 0]$ by a slow slope, as we can see in Figure 1.7 and formula below:

$$f(x) = \begin{cases} x \quad for \quad x \geq 0 \\ ax \quad for \quad otherwise \end{cases} \qquad for a \leq 1 \tag{1.17}$$

LReLU was introduced to fix the problem in ReLU. Instead of the function outputting a zero when x is smaller than zero a small negative slope is used. The motivation for using LReLU instead of ReLU is that constant zero gradients can also result in slow learning, similar to a saturated neuron with a sigmoid activation function. Furthermore, some of the neurons may not even activate. However, the preceding effect sacrifices the zero-sparsity and according to the authors in [35], may obtain worse results than when the neurons are entirely deactivated. In figure 1.7, sigmoid, Tanh, ReLU and LReLU are shown respectively. We are using LReLU in our work as our activation function, since it has demonstrated the best performance out of all the above mentioned functions.

1.5.3 Pooling layer

A pooling function replaces the output of the network with a statistical summary of the nearby outputs. For example, max-pooling which is extremely useful in image restoration problems gives us the maximum output within the neighborhood.

We elaborate max pooling functionality by the aid of Figure 1.8. Figure 1.8 is a 4×4 matrix which is converted to a 2×2 matrix after max pooling. This process which is called downsampling and has been commonly used in discrete time systems[39]. After a signal conversion from continuous to discrete domain, the effective sampling rate can be reduced via max-pooling. Max-pooling leads to faster convergence rate by selecting superior invariant features which improves generalization performance[39].

Max pooling is an operation that is typically added to CNN's following the individual convolutional layers. When we integrate a model with max pooling, it reduces the dimensionality of images by reducing the number of pixels in the output from the previous convolutional layer. The output of the max-pooling layer is given by the maximum activation over non-overlapping rectangular regions.

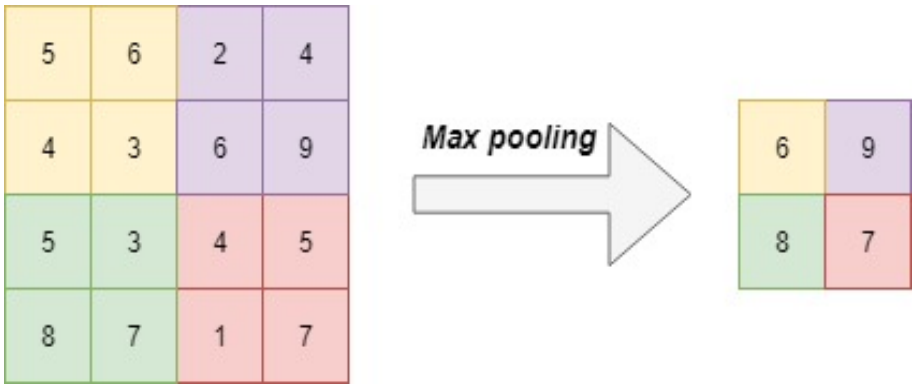


FIGURE 1.8: In each window of 2×2 the bigest data is chosen as the representetive of that window, in The first window colored yellow the highest number is 6, so 6 represents that window

1.5.4 Data types

The data that we use in CNN usually have several channels, and each channel is an observation of a variant quantity. The table below shows some examples of different data types [18].

	Single-channel	Multi-channel
1-D	Audio waveform	Skeleton animation data
2-D	Audio data that has been preprocessed with a Fouriet Transform	Color image data
3-D	Volumetric data, Medical imaging such as CT scans	Color video data

TABLE 1.1: Example of different data types

The data that we are working with is 3-D multi-channel type data, i.e. that is a collection of 2-D image data. *Voxel* is a data type tailored for our application which is described further in the following subsection.

Voxel

In 3-D raster graphics, the volume is divided into evenly spaced rows and columns, covering the three different directions (up-down, left-right, in-out). In other words, 3-D space can be partitioned into cubes, also known as voxels (volume elements).

A voxel is a unit of information, same as a pixel for 2-D; voxel defines a point in three-dimensional space. In a voxel reperesentation each 3-D point consist of a position, color, and density. With this information and 3-D rendering software, a two-dimensional view from various angles of an image can be obtained and processed by a computer[49]. For X-ray, CT, and Magnetic Resonance Imaging (MRI) scans, the widely used data type is a voxel.

1.6 Contribution

The contribution of this work is examining three main approaches towards denoising abdominal LDCT, as summarized in the following:

- Residual Encoder Decoder Convolutional Neural Network (RED-CNN) [24]
- Generative Adversarial Network (GAN)
- Our proposed framework, which is a combination of the latter two approaches with modifications adopted to best tailor it for the application of our interest

In the following chapter there is a review of previous works on denoising natural and medical images, followed by Chapter 3 and 4 which provide descriptions of how state of art techniques are implemented to solve the given problem with a discussion of their achieved results. In chapter 5 we go through our proposed approach which is a novel alteration and combination of Residual Encoder-Decoder Convolutional Neural Network

(RED-CNN) and Generative Adversarial Network (GAN), calling it RES-GAN. The last chapter concludes by a discussion and overview of this thesis and its future works.

Chapter 2

Literature Review

In Chapter 1 we briefly discussed the problem of denoising LDCT images and the method that we deploy to solve it, which involves using a deep neural network. In this chapter, we review the works of other researchers on this problem. To better understand the concept of image denoising, first we revisit some previous works on natural image denoising in section 2.1, and following that in section 2.2 a review on medical image denoising, which is mostly based on the former denoising methods.

The two main artifacts in images are blur and noise. Blur is innate to image acquisition systems because digital images have a finite number of samples and must respect the Shannon-Nyquist sampling conditions[48]. The meaning of noise is unwanted signal; unwanted electrical fluctuations are also called noise. A pixel value is light intensity measurement, usually made by Charge Coupled Device (CCD) or CMOS sensors connected to a light focusing system. Each detainer of the image sensor is a square, and the incoming photons are being counted. If the light source stays constant, the number of photons received by each pixel stays around its average as a result of the central limit theorem. Furthermore, if each captor is not cooled enough, it generates spurious heat photons. The effect is called "dark noise" [4].

Image denoising is a common issue in various imaging systems. Therefore, there

are diverse existing methods for solving it. The important property of a good image denoising approach is the ability to completely remove noise as far as possible while it preserve the edges [43].

2.1 Natural Image Denosing

In natural image denoising, Zhang et al. [52] compare several conventional methods including Block Matching and 3-D filtering (BM3D) [12], Last Setting for Learned Simultaneous Sparse Coding (LSSC) [36], Nonlocally Centralized Sparse Representation (NCSR) [15] and, Weighted Nuclear Norm Minimization (WNNM) [20] as well as CNN based methods. Among those, BM3D shows the best performance in term of quantitative measurement, and between CNN based methods, residual learning demonstrates the superior performance over full image learning with CNN. Some of these commonly used denoising methods are summarized as follow:

1. BM3D: This method is based on sparse representation in the transform domain. In a procedure called collaborative filtering similar 2-D images blocks are grouped into 3-D data arrays [12]. There are three steps for this method, including the 3-D transformation of a group, shrinkage of the transform spectrum, and inversing 3-D transformation. Applying these three steps progressively results in a 3-D estimation of jointly grouped image blocks.

2. LSSC: This method proposes sparse coding as a framework for combining non-local means and sparse coding approaches [36].

3. NCSR : Dong et al. [15] present a model for image restoration, which defines the difference between the sparse code of the noisy image and unknown original image

as sparse coding noise. The difference should be minimized and used to enhance the performance of sparsity-based image restoration.

4. WNNM: The standard nuclear norm minimization regularizes each singular value equally to pursue the convexity of a low rank matrix factorization problem. Gu et al. [20] proposed a method that singular values assigned as weights and when the weights are in a non-increasing order, WNNM is still convex and presented the optimal analytical solution. On the other hand, when the weights are in an arbitrary order, they suggest an iterative algorithm to solve them. Finally, when the weights are in a non-decreasing order, they proved that the iterative algorithm can result in an analytical fixed point solution, which can be efficiently computed. They claim that their algorithm outperforms many state-of-the-art denoising algorithms such as BM3D in terms of both quantitative measure and visual perception quality.

To the best of our knowledge BM3D outperforms most of the existing methods for Additive White Gaussian Noise (AWGN) denoising, and if not all, it is still remains the most popular method for natural image denoising. However, its performance decreases as the noise level increases in images since it is harder to find a proper match for reference blocks in the presence of highly corrupted pixel values [22].

We discussed the advantages of the above methods nonetheless, they all suffer from the following drawbacks [52]:

- They mostly contain a complex optimization problem in the testing stage, therefore they are time-consuming.
- They are non-convex in general and involve manually parameter choosing.

Machine learning-based methods come in the picture to overcome quoted drawbacks.

In general, machine learning methods are divided into two main categories, including *supervised* and *unsupervised* learning. In supervised learning, a model has conferred with an input feature x and labeled pairs y. They can take several forms, depending on the learning task. In a classification setting, y is generally a scalar, while in regression, it could be a vector of continuous variables. Supervised training typically aggregates to find model parameters that best predict the data based on a loss function [34]. On the other hand, unsupervised learning deals with data-set without labels, and it is trained to find a pattern such as principal component analysis and clustering methods. An advantage to the unsupervised method is adapting to perform under many different loss functions [34]. Using machine leaning for image denoising benefits from the following advantages:

1. Enhancing efficiency and flexibility to recognize image characteristics
2. Being able to handle blind denoising with unknown noise level[52]
3. Eliminating the need for a pipeline of specialized and hand-crafted methods by being able to learn from images

Later in Chapter 3 we explain the details of residual convolution and deconvolution layers in a network. However, Mao, Shen, and Yang [37] were the first who use this method in their work. They used convolutional layers for feature extraction and deconvolutional layers for recovering the image detail. The convolutional layers deploy the network to capture the image details and removing the noise. They proposed to link the convolutional and deconvolutional layers with skip-layer connections. Skip connections pass every convolutional layers to their mirrored deconvolutional ones. Adopting skip-layer connections provides benefits, such as ability of a signal to be back-propagated directly to lower layers. It also helps with gradient vanishing problem and makes the training of the deep networks more prosperous. Another aspect of using skip-layer is that

skip connections move image details from convolutional layers to deconvolutional ones, make recovering the original images much easier. All in all result in gaining restoration performance.

Zhang et al. [52] separate noise from a noisy image, using feed forward Denoising Convolutional Neural Network (DnCNN). Instead of outputting the denoised image $\hat{x}$ from Equation 1.10 directly, DnCNN is designed to predict $\hat{v}$, where v is the difference between ground truth High-Resolution (HR) image and bicubic up-sampling of Low-Resolution (LR) image. Consequently, the problem becomes a Single Image Super Resolution (SISR) problem. However, in SISR the noise v is different from AWGN. They also claim that their method solves all the image denoising problems including, SISR, JPEG deblocking, and Gaussian denoising.

2.2 Medical Image Denoising

There are three leading solutions for filtering the noises caused by lowering the projecting X-ray, including sinogram filtering before reconstruction, iterative reconstruction, and image post-processing after restoration. In this section, we are going to review each of these methods.

2.2.1 Sinogram filtering

Sinogram filtering works with either raw data or log-transformed data before image reconstruction, such as Filtered Back Projection(FBP). An advantage of using this method is that the noise characteristics are well known. Wang et al. [45] discussed the fact that LDCT sinogram data have been shown to be signal-dependent with an analytical relationship between the sample mean and sample variance. They have used this dependency

in order to propose a novel filtering method. The method they proposed is not based on nonlinear filters' statistics. Their system shows drastically better results in terms of noise suppression and structure preservation compared to the previous works on sinogram filtering.

According to [45], there are two main edge-preserving filtering methods; Anisotropic diffusion filtering and Non-Linear Gaussian filtering Chain (NLGC). In the former approach, edges are detected in multiscale space, and the diffusion strength is controlled by gradient image intensity. This method is popular in medical imaging, while the drawback is that it does not provide image dependency guidance for selecting an optimum gradient magnitude. Therefore, lots of works have been done to overcome this limitation.

In the latter method, NLGC, there are some advantages including, none critical choice of parameters and, being faster than robust statistic estimation smoothing. Moreover, three to five filters is enough to separate the random noise from the structure of an image. The contribution of works before [45] is changing the formula for finding a noise level estimation value. Their proposed method is useful for noise reduction compared to previous edge-preserving noise smoothing methods. Methods before [45] are not optimal choices for LDCT noisy sinogram since they can not remove the streak artifacts in the LDCT image.

There are some drawbacks to sinogram filtering method, including scarcity of raw data; most of the time, it is hard to have access to raw data. Moreover, the sinogram filtering methods often suffer from spatial resolution loss when edges are not well preserved [10].

2.2.2 Iteration reconstruction

For any analytical reconstruction algorithm, like FBP the measurement process and the projection data are given by continuous functions [16]. The iterative reconstruction method is a combination of statistical properties of data in the sinogram domain and information in the image domain.

The emerging of computers resulted in using a variety of iterative reconstruction algorithms and their application for CT imaging. The iterative reconstruction implementation reduces the CT image noise that drives to improving the image quality, compared to those using FBP reconstructions.

Work proposed by Zhang et al. [16] make use of iterative reconstruction in which they have access to both raw data and CT images. Their method has been proved to be a powerful tool for noise reduction. These types of methodology generally has resulted in superior performance in comparison to sinogram filtering.

Using a Total Variation (TV) as a regularization term is the most common term for disposing of the noise. The drawback to TV is causing smoothness edges and structures, called blocky effect, which can have decisive clinical value. Zhang et al. [54] establish a new TV method called fractional order for defeating the blocky effect. Their method achieve better results than other TV methods on detail and contrast preservation. They compute characteristics of each pixel with a local variance, which is a statistical variable defined as local variance and introduced into the computation of the fractional order. We can not deny that sinogram filtering and iteration reconstruction are based on the processing of raw data. Having access to raw-data is not always possible, and working with their structure is time demanding. That is where post-processing methods, which only based on image processing were introduced.

2.2.3 Post-processing after reconstruction

Earlier we mentioned that a considerable disadvantage to sinogram filtering and iterative reconstruction methods is the scarcity of raw data. The advantage of post-processing methods over the other two is ability to perform without having access to raw-data.

Post-processing of images with noise reduction filters can decrease noise in LDCT images. A method made in [27] using six different filters results in an outstanding noise reduction methodology in LDCT images. Their method eliminates appearance of the noise without perceptible loss of definition of anatomic structures. The fundamental processing steps of noise reduction in their work is presented in Figure 2.1.

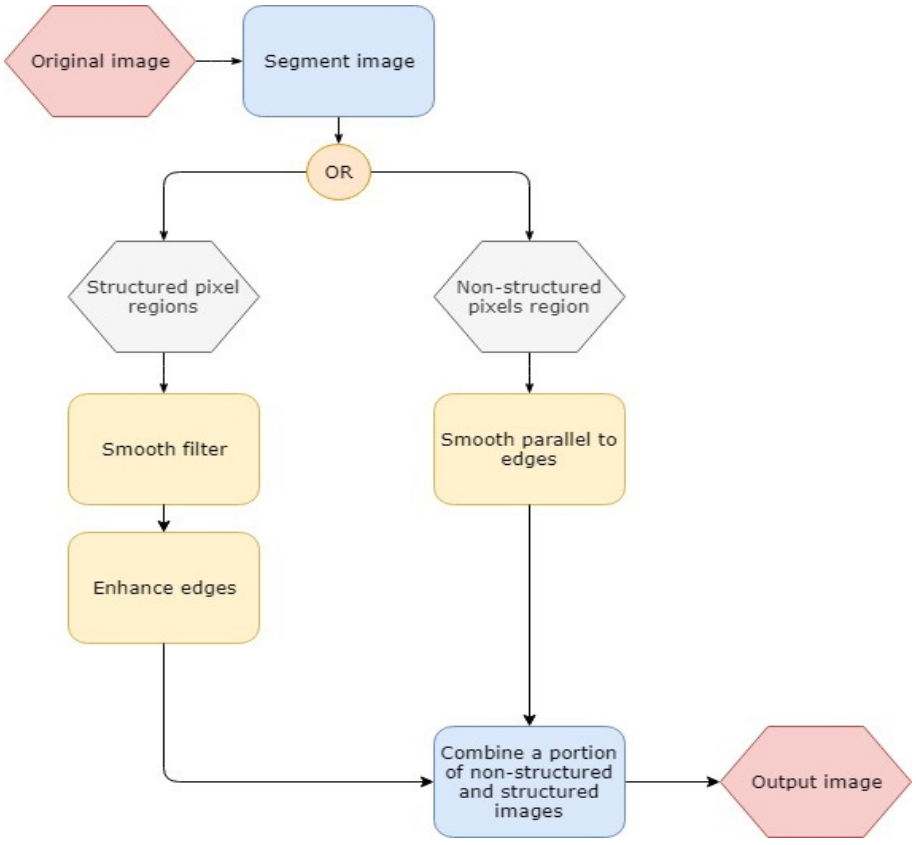


FIGURE 2.1: A picture of the fundamental processing steps of noise reduction in [27]

According to Figure 2.1, they are using gradient analysis methods to separate the image into structured and nonstructured regions. A threshold parameter controls the segmentation process. The nonstructured regions are filtered with a low-pass filter while the structured regions are directionally filtered with a smoothing filter operating parallel to the edges with an enhancing filter operating perpendicular to the edges. A blending parameter regulate the recombination of the structured and nonstructured segments.

Majority of state-of-art approaches in the field of LDCT image denoising, deploy Convolutional Neural Networks (CNN), which have demonstrated promising performance as a result of their capability of learning high-level noise characteristics.

Chen et al. [8] provide an extensive survey on several frameworks involving denoising, comparing the performance of noise dependence methods such as median filtering, Gaussian filtering, anisotropic diffusion, etc. which are designed for a particular noise type. On the other hand, approaches based on machine learning tend to provide a general solution to any noise model. ML methods are capable of learning more complicated noise features, mainly because they can be exposed to a considerable amount of training samples. According to their work [8] Sparse Representation(SR) is popular for image processing. The main idea of SR is to extract patches of an image with a pre-trained dictionary. The dictionary can be divided into two groups, including analytic dictionaries such as Discrete Cosine Transfrom (DCT) wavelet, Fast Fourier Transform (FFT), etc. and learned dictionaries which conserve application if they have proper training samples.

Burger, Schuler, and Harmeling [5] show that CNN tends to have better quantitive and perceptual results compared to the non-learning based method such as BM3D developed by [12] if the capacity of ML is chosen large enough. Large in terms of the number of hidden layers and the patch size to contain adequate information.

Residual deep learning is another class of approach to implement LDCT image denoising. In [10] they are using a combination of the auto encoder, deconvolutional network, and shortcut connections. Their proposed method helps to achieve outstanding performance in low-dose CT image denoising. To define the problem Chen et al. [10] use an FBP reconstruction from LD scan and starting to work with reconstructed CT image, adopting the fact that deep learning based methods are independent of the statistical distribution of image noise level. Works before [10], have used CNN, which is suitable for image restoration, but as a result of multiple down-sampling, some image details can be removed. However, a residual network inspired by the work of [37], where auto-encoder and CNN are merged, solved the problem. Instead of using fully-connected layers for encoding and decoding, Mao, Shen, and Yang [37] used both convolutional and deconvolutional layers. The other change they made to the network is removing the Rectified Linear Units (ReLU) layers before summation and adding shortcuts to improve the learning process.

In medical image denoising, methods based on CNN, transfer learning [44] and Generative Adversarial Networks (GANs) are starting to be methods of interest. According to [44], when solving the problem with small training data-set, using transfer learning is so efficient. Transfer learning definition is using a pre-trained network as a feature extractor and fine tuning that pre-trained network on our desirable data-set.

GANs were first introduced by Goodfellow [19] primarily to secure networks against adversarial attacks. However, GAN's functionality has been expanded to other applications, including image restoration. The general GAN structure consists of two networks; *Generator* and *Discriminator*. The role of the generative network is to produce fake data, while the discriminator is deployed to distinguish real data from fake. An architecture of GAN is shown in Figure 2.2. By training the latter, two networks simultaneously, can form a generative model which closely maps scenarios from the real world [19].

Wolterink et al. [50] achieve exceptional performance, despite the misalignment of LDCT and NDCT. However, the drawback introduced by approaches based on GANs includes the complexity involved with the training of the two counterparts of these networks. They use three different loss componenets for training GAN networks, only voxel-wise loss, voxel-wise plus adversarial loss and finally, using only adversarial loss. Previous methods [9], [28], [29] minimize the per-voxel squared error between ground truth NDCT and LDCT. Minimizing the squared error results in smoothed images that lack the structure of an NDCT image. These smoothed images are the reason why they conclude that voxel-wise regression alone in noise reduction is inadequate.

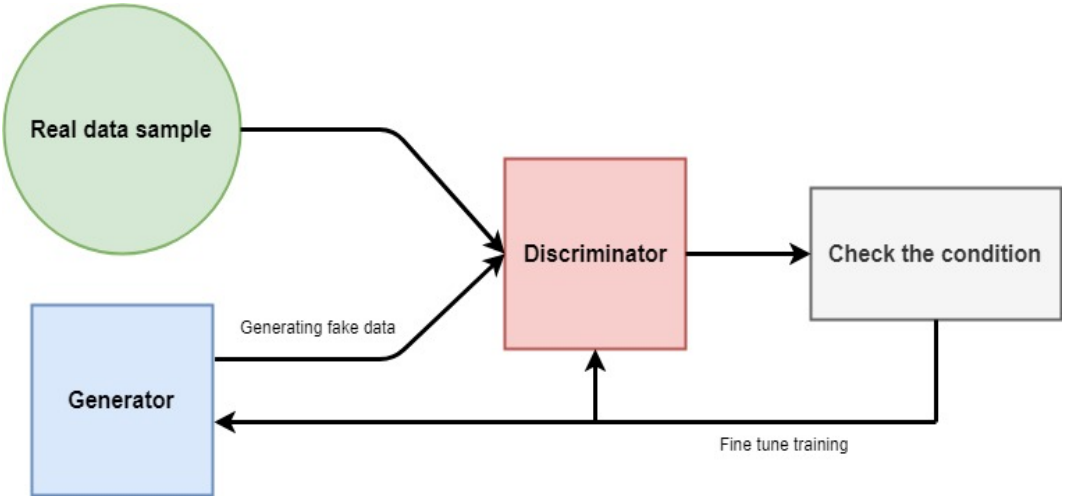


FIGURE 2.2: Architecture of a GAN

Consequently, they proposed training a generator CNN along with an adversarial discriminator CNN as their network. The performance of discriminator is a loss term for optimizing the generator, and results in getting more realistic image estimations from the generator.

They proposed two ways for comparing images generated by the generator and the ground truth NDCT. The first one is minimizing the voxel-wise error between generated CT and NDCT. The second one is a discriminator loss which is trained simultaneously to differentiate between normal and low dose CT. If the discriminator can identify NDCT

easily, the generator has to improve its production. Equation below shows the problem:

$$I_{LD} = I_{ND} + N \tag{2.1}$$

In which I_{LD} is a patch of low-dose image and I_{ND} is corresponding patch of NDCT image, using as the reference for training. N is the noise added to NDCT. This equation is exactly the same as Equation 1.8.

In [50] all layers use leaky rectified linear activation function (LReLUs) except last layer, for stability purpose. Loss components which affect generator are, first squared error between $G(I_{LD})$ and I_{ND}, second the ability of discriminator to identify generated images. A binary cross-entropy measures the second one. $G(I_{LD})$ is the output of the generator network with I_{LD} as input. They optimize generator in their work in three ways:

1. Voxel-wise loss between $G(I_{LD})$ and I_{ND}

2. Voxel-wise loss plus discriminator feedback

3. Only discriminator feedback

Two first proposes need spatial alignment between NDCT and LDCT, but the third one doesn't need alignment.

Wolterink et al. [50] achieve exceptional performance, despite the misalignment of LDCT and NDCT. However, the drawback introduced by approaches based on GANs includes the complexity involved with the training of the two counterparts of these networks.

In our work we have used both GANs and Residual CNN which both give acceptable results among the state-of-the-art methods, in Chapter 3 we discuss the residual learning

methods, and in Chapter 4 discuss and show results of GAN's method. In Chapter 5 we have our proposed method and the results of running tests.

Chapter 3

Residual Learning Approach

3.1 Introduction to Residual Learning

In this chapter, we discuss an eminent image denoising technique, known as residual deep learning. In a neural network with the network depth increasing, accuracy may get compromised due to network saturation. Residual learning approach aims to address the potential performance degradation caused by increasing the network depth. Deeper networks generally tend to have a smoother training process with residual mapping as opposed to original mapping.

Stacking multiple dense convolution layers, does not necessarily increase the learning rate. Although techniques such as employing more suited activation functions such as ReLU might help, the problem may still remain. Inspired by the work done by [24], we use a concept called *skip connections* which involves jumping over a number of layers. This technique allows us to increase the number of layers without hesitation. To summarize, two advantages are introduced as a result of leveraging this method [24]:

- Preventing degradation of training accuracy once traversing deeper into the network [24].

- Increasing the training pace.

Earlier in section 2.2.3, we discussed some recent work developed on residual learning. In the following section inspired by the work of [11], we define and remodel the denoising of LDCT problem with the use of residual neural networks.

3.1.1 Problem Description

Let $Y \in R^{m \times n}$ and $X \in R^{m \times n}$ denote an LDCT image and its corresponding NDCT counterpart, respectively. The following procedure aims at establishing a function, F, that maps Y to X:

$$F(Y) = X \tag{3.1}$$

The simplest form of mapping Y to X would be addition of a noise term, $Z \in R^{m \times n}$, as shown below:

$$Y = X + Z \tag{3.2}$$

The objective in residual learning approach is to learn the noise (residual) maps rather than the actual NDCT images. The obtained noise, Z, can then be subtracted from LDCT image Y to produce the NDCT denoised version X. In Figure 3.1 a typical layer of a residual network is shown in which $R(Y) \approx Z$ is subtracted from Y to produce $F(Y)$. This procedure constitutes *Residual Learning* of CNN. The input to the network is still a patch of LDCT, Y, and the output is $F(Y)$, a close estimate of NDCT patch, X.

For calculating $R(Y)$, we implement a network with ten layers, including five convolutional and five deconvolutional symmetric layers for feature extraction and reconstruction

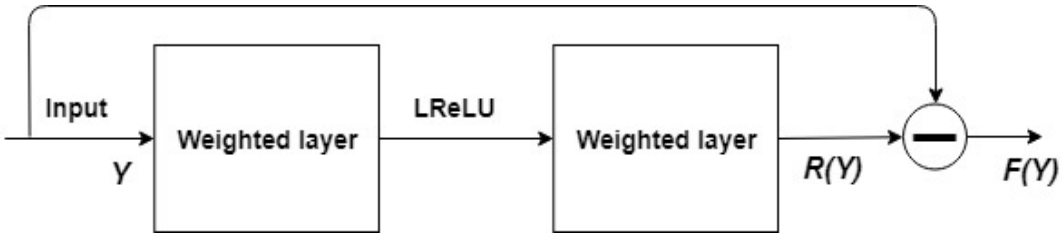


FIGURE 3.1: A structure of a Residual NN, the weighted layers are for learning the type of residual mapping: $F(Y) = Y - R(Y)$

of image details, respectively. After a preliminary analysis of various activation functions discussed in 1.5.2, LReLU activation function [41] is deployed for each layer.

3.2 Network Architecture

The architecture we are adopting in this portion of our implementations is a form of an autoencoding scheme. Traditionally auto-encoding is considered to be a data compression algorithm, for specific applications, i.e., it can only compress data similar to what it has been trained with. Although the autoencoder approach was first introduced for unsupervised learning, it is also sufficient for image restoration purposes, including image denoising.

What follows in the upcoming subsections is a discussion of the fundamental segments of the autoencoder design executed in the scope of this chapter which include: patch coding, feature extraction, structure recovery, and skip connection.

3.2.1 Patch Coding

Image denoising schemes are implemented in mainly two classes of: pixel-based image filtering and patch-based image filtering. In the former approach, any processing is done at the pixel level and the similarity of each pixel with all its neighbouring pixels within

a given window is calculated. The left box in figure 3.2 represents a pixel-based method. However with the patch-based denoising, the noisy image is divided into blocks and the proximity function is now calculated between patches. The box on the right in figure 3.2, shows a patch-based search within a certain window size and similar/close patches to the reference patch of our interest are circled.

Lack of sufficient labelled training data has always been a problem in the medical image domain. In field of medical image restoration, patch coding is commonly used in order to increase the number of samples. In [10], a patch coding technique is proposed

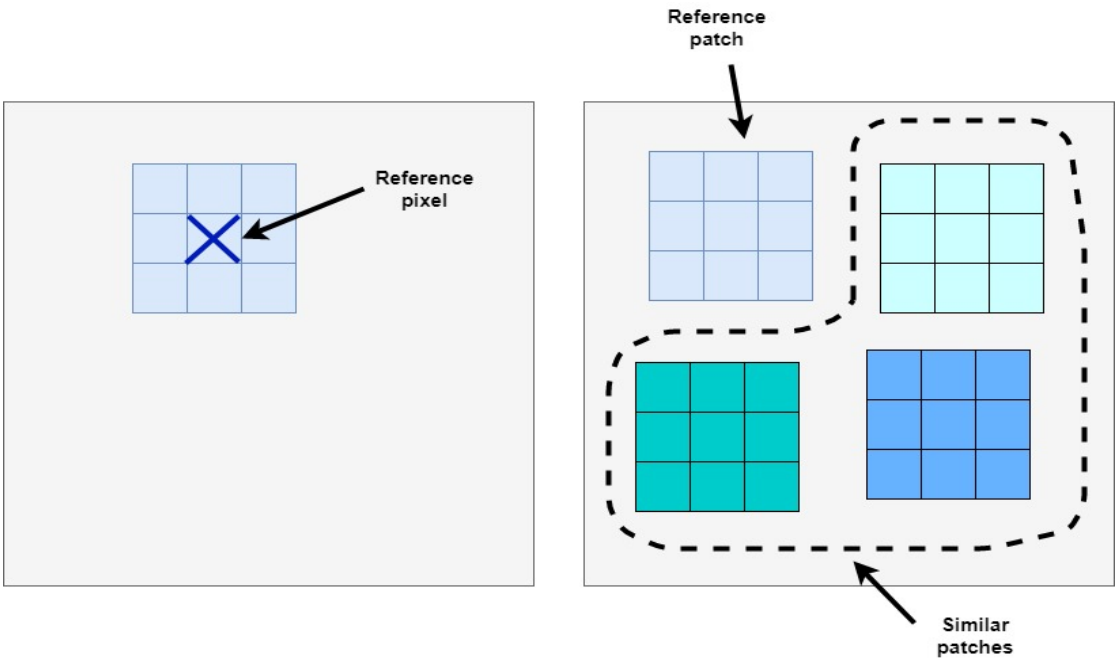


FIGURE 3.2: The left box respresents a pixel-based is a $N \times N$ size window with a reference pixel in the center, and neighboring pixels are the 8 pixels sharing a boundary with it, representing pixel-based method; Right window of the same size, shows patch coding method, where a reference patch and similar patches are shown respectively

in which the patches are overlapping, and subsequently, perceptual differences of local regions are detected. In our work, we extract fixed-size samples which correspond to

patches from LDCT images and their corresponding NDCT counterparts. In our network, we extract patches from the training image with fixed slide size of 64×64, and the result is shown in section 3.3.

3.2.2 Feature Extraction

Feature extraction procedure includes derivation of features that best describe the relevant information contained in a pattern in order to increase the feasibility of its following classification process. In terms of feature extraction, the extraction of suitable features is always a topic of concern. It is often crucial to discriminate between the relevant and irrelevant features. In image processing, feature extraction is a particular form of dimensionality reduction. The main goal of feature extraction is to obtain the most relevant information from the original data and represent that information in a lower dimensionality space [33].

Features of an image can be extracted by its content: either in form of color, texture, shape, position, dominant edges of image items or regions. To extract local features, traditional methods for example calculate the first and second-order derivative of the image patch. The approach is similar to filtering the image with high-pass filters. Deep learning-based methods automatically learn these filters from training data. In other words, the patches from NDCT images are our labeled training dataset that network tries to learn features from. In this network, we deploy fully-connected convolutional layers for feature extraction, consisting of 5 layers to suppress image noise and artifacts. In Chapter 1, we discussed the pooling layer and its advantages, but the major drawback in using pooling layer is discarding important structural details [10]. Therefore, it is removed from the portion of our implementation.

To map image patches into a feature space, we use stacked encoders with Equation 3.3 [46]:

$$F_l = LReLU(W_l * F_{l-1} + b_l) \tag{3.3}$$

In Equation 3.3, W_l and b_l denote the convolutional filter and the bias of the l_{th} layer, consecutively. F_l demonstrate the output feature map and F_0 is the original LDCT with fixed block size. LReLU is a non-linear mapping, using for activation function as previously discussed in section 1.5.2. The following points reiterate the motivation behind using LReLU as activation function:

- The greatest advantage of using a LReLU or ReLU is indeed non-saturation of its gradient, which greatly accelerates the convergence of stochastic gradient descent compared to the sigmoid or tanh functions.
- Compared to tanh and sigmoid neurons that involve expensive operations, the ReLU can be implemented by simply thresholding a matrix of activations at zero.

3.2.3 Structure Recovery

In this step of our work, we review the details of deconvolving the features extraction step, for reconstructing the denoised image. Removing pooling layer from convolutional layers helps retaining some important structures, but we still lose some crucial details after a series of convolutional layers. To overcome this problem, we use deconvolutional layers for recovery of structural details [10, 37]. For deconvolutional layers Equation 3.4 can be used to describe the procedure, with $W_l\prime$ and $b_l{}'$ denoting weights and biases of deconvolutional layer respectively.

$$D_l = LReLU(W_l' \bigotimes D_{l-1} + b_l') \tag{3.4}$$

There is the need to ensure that convolutional and deconvolutional layers match exactly in terms of numbers as well as filter size.

3.2.4 Skip Connection

As the network gets deeper, the gradient vanishing problem tends to affect our network more, hence compromise the training procedure. The use of LReLU activation function was a measure to avoid this problem. Additionally as demonstrated by the work done by [10, 37], we introduce the *skip connection* technique to our network. The skip connection technique involves jumping some layers in the neural network, and feeding the output of one layer to another layer without undergoing any processing.

Usually, some information is captured in an initial layer and is required for reconstruction in a following layer. If we do not use the skip connection architecture, there would be a risk of losing information. Consequently, a piece of information that we have in the first layers can be fed explicitly to the later layers using this method. The following summarizes the benefits we are gaining by leveraging the skip connection shceme:

- Being able to pass image details forwardly and recovering clean image
- Helping pass the gradient backwardly, resulting in finding better local loss function minimum

3.2.5 Data-set

The training and test data-set we use is the data-set presented within a coding challenge held in 2016, known as *"NIH-AAPM-Mayo Clinic Low Dose CT Grand Challenge"*. The data-set consists of contrast-enhanced abdominal CT examination taken from 10 patients. In this data-set, both normal-dose and quarter dose CT fan-beam reconstruction counterparts are included. A method proposed by Chen et al. and [7] was used to decode the sinogram data into images. We are using the produced images to construct our data-set. Table 3.1 summarized our data-set in terms of patient ID, the number of slices for each CT, and normal/routine-dose and low/quarter-dose tube current in milliampere. Each slice of CT consists of 512×512 pixels.

Patient ID	Number of slices	ND tube current(mA)	LD tube current(mA)
L067	310	234.1	59.2
L096	500	327.6	82.9
L109	254	322.3	79.2
L143	418	416.9	105.5
L192	370	431.6	109.2
L286	300	328.9	82.2
L291	450	322.7	81.7
L310	340	300.0	73.7
L333	400	348.7	88.2
L506	300	277.7	70.2

TABLE 3.1: Data set

3.3 Experimental Result

In this section we discuss a brief overview of widely used image evaluation techniques. We evaluate the performance of our implementation using residual learning method on different combinations of training and test data-set, while varying the size of patches. In our experiments, we consider both patch size 64×64 and 55×55. However, patch

size of 64×64 yielded a better performance in terms of image qualitative measurements. Therefore, only the results for 64×64 patches are presented.

3.3.1 Image Evaluation

Image quality evaluation methods are mainly categorized into objective and subjective methods. Subjective methods are based on human opinion and are conducted with often no need to the reference. On the other hand, objective methods are based on comparisons using explicit numerical criteria [25]. Due to high cost of subjective methods, we require accurate objective metrics for making comparisons.

Objective methods are classified into three types:

1. Full-reference: most existing approaches are known to be full-reference. In this category, the original image is available as a reference for the impaired one.

2. No-reference or blind: the reference image is not available.

3. Reduced-reference: some attributes of the reference image is available.

In this work, there exists the luxury of access to Full Dose CT images. Therefore, as our objective evaluation method, we may use full-reference quality metrics. In this subsection, a number of full-reference metrics are described which are potentially suitable for our work. The simplest and most used full-reference quality measurement is Mean Squared Error (MSE) [47]. For a refernce image f, and a test image g, both size $M \times N$ we define MSE as follow:

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [f(i,j) - g(i,j)]^2 \tag{3.5}$$

And based on MSE, Peak to Signal Noise Ratio (PSNR) is:

$$PSNR = 10\log_{10}\frac{{MAX_I}^2}{MSE} \tag{3.6}$$

Where MAX_I is maximum possible pixel value of the image. The common used equation for PSNR is:

$$PSNR = 20\log_{10}(\frac{MAX_I}{\sqrt{MSE}}) \tag{3.7}$$

This equation shows that if MSE reaches 0, then the PSNR approaches to ∞, meaning higher PSNR corresponds to higher image quality [25].

Another famous full-reference quality metric is Structural Similarity Index Method (SSIM). SSIM is designed by modeling any image distortion as a combination of three factors including, loss of correlation, luminance distortion and contrast distortion. It is defined as:

$$SSIM = l \times c \times s \tag{3.8}$$

Where each of l, c, s are luminance comparison function, contrast comparison function and, structure comparison function respectively. The luminance comparison function measures the closeness of the two images define as:

$$l(f,g) = \frac{2\mu_f\mu_g + C_1}{\mu_f^2\mu_g^2 + C_1} \tag{3.9}$$

In this equation μ_f and μ_g are mean luminance of images f and g. The $l(f,g)$ only equals to one if $\mu_f = \mu_g$. Second factor which measures the closeness of contrast between image f and g is:

$$c(f,g) = \frac{2\sigma_f\sigma_g + C_2}{\sigma_f^2\sigma_g^2 + C_2} \tag{3.10}$$

The last one which is structure comparison function, measures the correlation coefficient between the two images f and g with covariance $\sigma_f \sigma_g$ between f and, g.

$$s(f,g) = \frac{\sigma_{fg} + C_3}{\sigma_f \sigma_g + C_3} \tag{3.11}$$

The positive constants C_1 C_2 and C_3 are used to avoid a null denominator. The positive value of SSIM is between $0,1$ and 1 is for when images f and g are exactly the same [25]. Further information of how to measure μ and σ can be found in [25].

Our evaluation approaches to compare our method with previous works in this area are PSNR and SSIM. In Chapter 4, we define another evaluation method, which is the closest to a subjective method, i.e., human perception.

3.3.2 Training and Test

In the learning of the mapping from LDCT to NDCT needs to estimate the weights of the convolutional and deconvolutional filters are determined. This is achieved by minimizing the MSE between the output of the network and the ground truth. We randomly extracted pairs of image patches from all of our CT training data-set images of all patients in 3.1 except the one we wanted to test with. We used cross-validation. Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called k that refers to the number of groups. The process is often called k-fold cross-validation when a specific value for k is chosen for our data-set. We have the abdominal CT of 10 patients, therefore, $k = 10$.

Cross-validation is popular since it is simple to understand. Another reason is that it generally results in a less biased or less optimistic estimate of the model's ability than other methods, such as a simple train/test split [26].

	LDCT	RED-CNN [10]	Proposed Residual CNN
PSNR	34.3094	39.1959	40.0227
SSIM	0.8276	0.9339	0.9473

TABLE 3.2: PSNR and SSIM of LDCT, RED-CNN [10], and our proposed residual learning

The general steps of cross-validation in our work are as follow:

1. Shuffle the data-set randomly.
2. Split the data-set into 10 groups.
3. For each unique group, we take a group as a hold out or test data-set and we take the remaining groups as a training data-set.
4. Fit our model residual learning on the training set and evaluate it on the test set.
5. Keep the evaluation score, which is PSNR and SSIM and discard the model.
6. Repeat steps 3 to 5 on a different set.
7. Using evaluation scores to summarize the capability of the model.

The quantitative results for the image data-set using cross-validation are shown in table 3.2. As can be seen from the table our implementation achieves PSNR and SSIM values that are slightly better than the one represented in [10]. Overall, our residual CNN is successful in generating NDCT from LDCT.

We implemented and trained our network using Adam optimizer with learning rate of 10^{-5}. the base learning rate for all layers are the same. The residual CNN is initialized to 0 for all biases and LReLU is used for activation function. Table 3.3 shows the quantity and other details of this network in particular.

Parameters in the proposed network	
Training environment	Specification
Patch size	64×64
Initializer	Random Gaussian distribution (0, 0.01)
Learning rate	10^{-5}
Number of iteration	2000
Optimizer	Adam
Loss function	MSE

TABLE 3.3: Hyper-parameters for Residual learning network

Figure 3.3 shows the loss function of our proposed network for one of the training-test samples. In essence, the loss function graph represents how much a certain quantity of difference between an estimator $F(Y)$ in Equation 3.1 and actual value X will be penalized by the model. The plot shows that the training process converged well. The plot for loss is smooth after 600 iteration, and the loss function approaches zero.

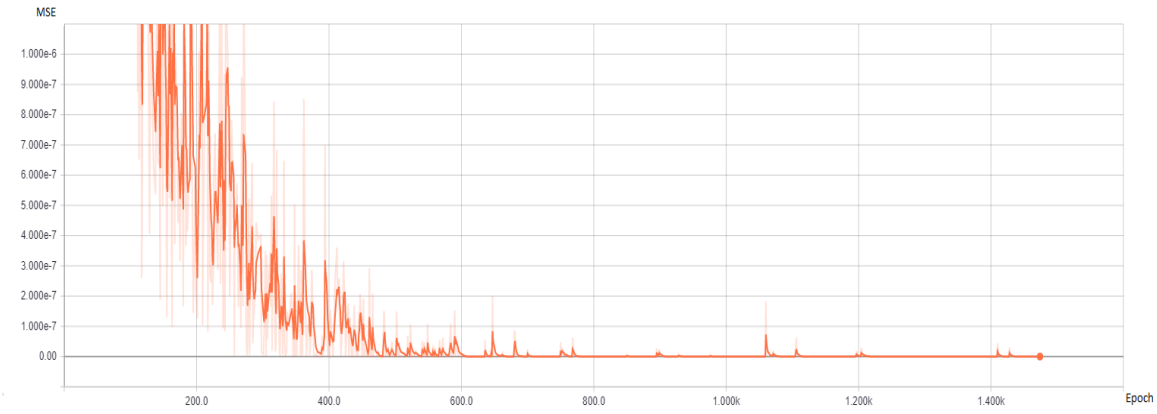
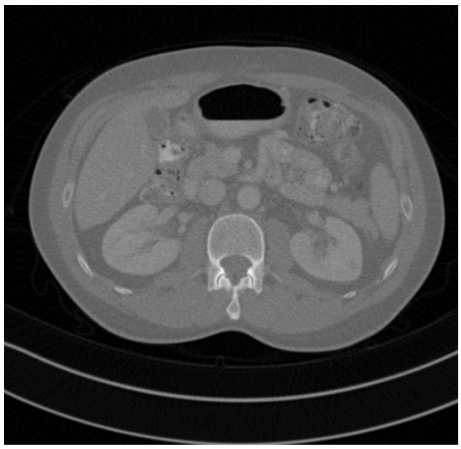
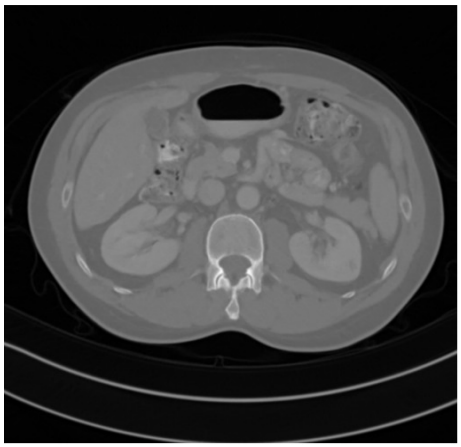


FIGURE 3.3: Loss function which is MSE, approaching zero after 600 iterations and settliing

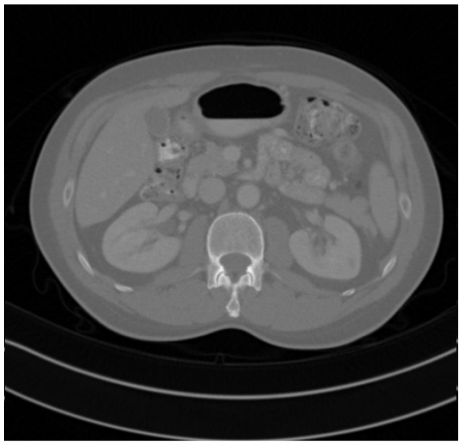
In Figure 3.4, a slice of abdominal CT in quarter-dose, final output, and full-dose/ground truth is shown from top to bottom, respectively. We can visually observe that Result of our proposed residual network is less fuzzy than the quarter dose, and we cannot perceptual tell the difference between NDCT and Result of our network. By all means, our network is successful in producing a denoised CT image from LDCT one.



Quarter Dose/ LDCT



Full-dose/ NDCT



Result

FIGURE 3.4: From top to bottom, Quarter-Dose/LDCT, Full Dose/NDCT and Result which is output of proposed residual network; All three images are from a same slice of L067 patient's abdominal CT

Chapter 4

GAN approach

In this chapter, we discuss another approach towards denoising LDCT images, called Generative Adversarial Network (GAN). We reviewed some previous works on GANs in Chapter 2. First, we discuss the detailed practice of GANs, and then we elaborate on our novel proposed network called Wasserstein Generative Adversarial Network (WGAN) with detail. In the end, we discuss our proposed approach and the results of evaluations.

4.1 Generative Adversarial Network

Goodfellow et al. [19] were the first who proposed GANs in 2014. GAN is a class of machine learning systems that has attracted attention recently. One of the outstanding merits of GANs is that they generate data that is similar to real data, which results in having multiple applications in the real world. They can generate images, text, audio, and video that is indistinguishable from real data. Images generated by GANs have applications in marketing, e-commerce, games, advertisements, and many other industries. Moreover, they have shown promising results for LDCT image denoising in terms of both quantitative measurement and visual perception.

As we raised earlier in 2.2.3, Generative Adversarial Networks consist of two models, *Generator* and *Discriminator*. The generator produces synthetic data and the discriminator, acting as a critic, tries to discover if the received data is either a sample of fake or real data [19].

We train both networks in alternating steps and lock them into a fierce competition to improve themselves. Eventually, the discriminator identifies the tiny difference between the real and the generated data. Then the generator creates images that the discriminator cannot segregate them from real data. By training the discriminator and the generator networks simultaneously, the generator network eventually converges and is capable of producing a model similar to the ground truth; NDCT.

GANs learn the internal representations of data. As mentioned earlier, GANs can learn messy and complicated distributions of data, which can be helpful for many machine learning problems. Also, after the training step, we end up having a discriminator and a generator. The discriminator network plays a classifier role which its task is classifying the objects. The main disadvantages to GANs are, being hard to train and time-consuming. The function these networks try to optimize is a loss function that essentially has no closed-form, unlike standard loss functions such as log-loss or squared error. Thus, optimizing this loss function is very hard and requires several trial-and-error regarding the network structure and training protocol.

In LDCT denoising, we use GAN to reconstruct a normal dose CT with generator network. In the following section, we discuss how we utilized GAN in order to outperform the state-of-the-art LDCT image denoising methods.

4.1.1 GAN In LDCT Image Denoising

The input to the generator network, are LDCT images and we use NDCT ground truth images for training both generator and discriminator. Now we have a double feedback loop. The discriminator is in the loop with NDCT images, and the generator is in another loop with the discriminator. We have to hold the generator constant while we train the discriminator and vice versa.

Each of the networks should be trained against a static adversary, and both networks must have a similar skill level. For example, if the discriminator is far better than the generator, then it will return values close to 1. In other words it can discriminate real and fake data with a probability close to 1. However, if it is the other way around and the generator is better, then it will frequently apply weaknesses in the discriminator that leads to a large number of false negatives. During the training discriminator gets both NDCT and LDCT as inputs and determines whether the input is real or not, by calculating the probability of the input image being real.

Loss component for our discriminator is an adversarial goal to differentiate LDCT and NDCT correctly. The whole point of having a discriminator is, if we have two NDCT images with same PSNR and SSIM, the one we have produced with using a GAN is visually better than the one that is only used a CNN or any deep learning methods. The reason behind this is that we yet do not have quality measurements comparable to human vision. To summarize, the steps a GAN takes in our work are:

- First the generator takes an LDCT image and generate an image in next level.
- Generated image is then fed into the discriminator alongside a stream of images taken from the ground-truth NDCT image.

- The discriminator takes in both real and fake images and returns probabilities, values between 0 and 1, where 1 representing a prediction of authenticity and 0 representing image being synthetic.

- The generator error and discriminator error both are used for training generator.

When the models are both multilayer perceptrons, the adversarial modeling is simpler. According to [19], the generator essentially defines probability distribution p_g over data x and $p_z(z)$ is input noise variable; $G(z;\theta_g)$, $D(x;\theta_d)$ represents mappings of generator and discriminator to data space, respectively. We often define GAN's training as a min-max game which G wants to minimize V while D wants to maximize it [19]:

$$\min_G \max_D V(D,G) = E_{x\sim p_{data(x)}}[logD(x)] + E_{z\sim p_z(z)}[log(1-D(G(z)))] \tag{4.1}$$

Gnereative and discriminative networks learned jointly by the alternating gradient descent. In GAN the generator model's parameters is fixed and a single iteration of gradient descent is performed on the discriminator using the ground truth images. Then they switch sides; set the discriminator and train the generator for another single iteration. Both networks are trained in alternating steps until they reach the point that is satisfying, for example, a quantitative image measurement.

4.1.2 Our WGAN Implementation

Insted of cost function defined in 4.1, and inspired by work of Yang et al. [51], we use Wasserstein distance estimation to compare data distributions and training the GAN. In section 4.1.3, we discuss the reason behind using Wasserestein distance. Instead of using a discriminator to classify or predict the probability of generated images as being

real or fake, the WGAN changes or replaces the discriminator model with a critic that scores the realness or fakeness of a given image.

Based on Equation 4.1 the training of our proposed WGAN network would be to solve:

$$\min_{G}\max_{D} V(D,G) = A + B + C \tag{4.2}$$

$$A = -E_x[D(x)] \tag{4.3}$$

$$B = E_y[D(G(y))] \tag{4.4}$$

$$C = \lambda E_{\hat{x}}[(\| \bigtriangledown_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \tag{4.5}$$

In Equation 4.2 terms $A + B$ is Wasserestein distance estimation, and C is gradient penalty term for network regularization. $V(D,G)$ is adversarial loss. x is the NDCT (ground truth), and y is the LDCT, and $\hat{x}$ is uniformly sampled along straight lines connecting pairs of generated NDCT and ground truth NDCT images. Also, λ is a weighting parameter. Compared to original GAN 4.1, WGAN removes log functions. Specifically D and, G are trained alternatively by fixing one and updating the other.

Inspired by work of [51] we are also using a pre-trained network called VGG-19 as another term in our cost function. A pre-trained model, is a model that has been previously trained on a dataset and contains the weights and biases that represent the features of whichever dataset it was trained on. VGG-19 is a convolutional neural network that is trained on more than a million images from the ImageNet [6] database. Further in this chapter we explain the reasons behind using both Wasserestein distance and VGG-19 in our overall cost function. Therefore, there is a loss component called VGG loss, F

added to our cost function and define as follows:

$$F = E_{(x,y)}[\frac{1}{whd}\|VGG(G(y)) - VGG(x)\|_2^2] \tag{4.6}$$

Where w, h, d, are the width, height, and depth of the feature space. Therefore the new network training problem is to solve:

$$\min_G \max_D L_{WGAN}(D,G) = A + B + C + \lambda_1 F \tag{4.7}$$

We defined A, B, C, F in Equation 4.3 to 4.5. λ_1 is a weighting parameter for controling the trade-off between adversarial loss and VGG loss.

The WGAN architecture for both training and, test phase is shown in Figure 4.1. In this figure, LDCT, or y is the input of the generator and in the output we have $G(y)$, we send this $G(y)$ to both VGG and discriminator and they both gives us a loss component; M refers to discriminator loss, and N refers to VGG loss.

To summarize GAN network and WGAN network are different in the following aspects:

- The loss function; using Wasserestein distance in WGAN instead of MSE
- Adding a new pre-trained network, called VGG and use it for computing perceptual loss
- Compared to the original GAN, WGAN removes the log function in the losses and also drops the last sigmoid layer in the implementation of the discriminator [21]

We are going to discuss the changes and their advantages over previous works, but first we need to discuss a number of loss functions. In the following section we review Wasserestein distance and its influence on our network.

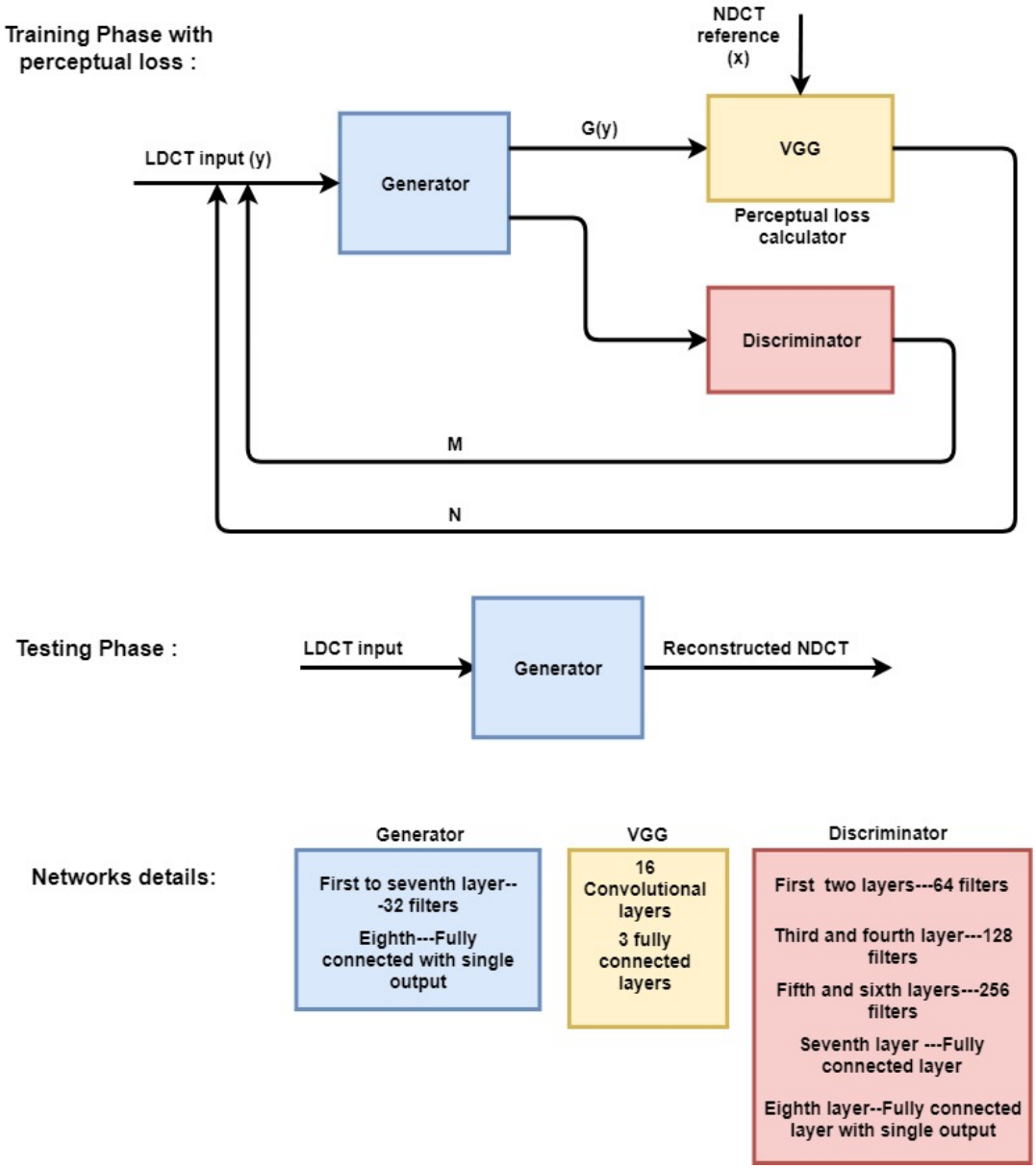


FIGURE 4.1: Training phase uses three networks to train the generator, but in test mode only generator is used and all of their networks details are shown in each corresponding frame

4.1.3 Wasserestein Distance

Most of the works in CT denoising domain, use Mean Squared Error, as their error between denoised CT image and the ground truth. Using MSE helps improving PSNR

but GANs specially WGANs tend to have better perceptual results. The perceptual loss stifle noise by comparing the perceptual features of a denoised output with ground truth in an established feature space. The GAN however, focuses more on drifting the data noise distribution from strong to weak. In the process of learning generative models, we assume that in data in hand comes from unknown distribution P_r. The purpose of using Wasserestein distance is to learn a distribution P_θ that approximates P_r, in which θ is distribution parameter[2]. In other words, the Wasserstein distance is the minimum cost of transporting mass in converting the data distribution P_θ to the data distribution P_r There are two methods for learning P_θ:

- Learn P_θ directly, where P_θ is some differentiable function.

$$P_\theta(x) \geq 0, \int_x P_\theta(x)dx = 1 \tag{4.8}$$

  We optimize P_θ through Maximum Likelihood Estimation (MLE). MLE objective function $MLE(\theta, x)$ in which data x independent and identically distributed for unknown population m is to find:

$$MLE(\theta; x) = \underset{\theta \in \Re^d}{max} \quad \frac{1}{m} \sum_{i=1}^{m} \quad logP_\theta(x^{(i)}) \tag{4.9}$$

  MLE estimation P_θ is equivalent to minimizing the Kullback–Leibler (KL) distance between P_θ and, P_r is defined as:

$$D_{KL}(P \parallel Q) = \int_{-\infty}^{\infty} \quad P(x) \quad log(\frac{P(x)}{Q(x)}) \tag{4.10}$$

  This equation is KL-divergence between two continuous distribution P and Q.

- Learn a function g_θ, able to convert an existing distribution z into P_θ where z is

often a simple distribution such as, uniform or Gaussian distribution.

$$P_\theta = g_\theta(z) \tag{4.11}$$

We can use both methods 4.8 and 4.11 for learning P_θ, however, the latter method is better because of the following reasons:

1. In Equation 4.10 if $Q(x) = 0$ where $P(x) > 0$ then the KL-divergence goes to $+\infty$ and it is not good for MLE.

2. Learning a $g_\theta(z)$ with a known distribution z, given a trained g_θ is very easy.

Training g_θ demands computing difference between the distributions. There are miscellaneous methods for computing distance between two continuous distribution P_r and P_g including:

- Total Variation (TV) :

$$\delta(P_r, P_g) = \sup_A |P_r(A) - P_g(A)| \tag{4.12}$$

- The KL divergence, mentioned in 4.10.

- Jenson-Shannon (JS) divergence:

$$JS(P_r, P_g) = \frac{1}{2} D_{KL}(P_r \parallel P_m) + \frac{1}{2} D_{KL}(P_g \parallel P_m) \tag{4.13}$$

where $m = \frac{P_r}{2} + \frac{P_g}{2}$

- Earth Mover (EM)/Wasserestein distance:

$$W(P_r, P_g) = \inf_{\gamma \in \prod(P_r, P_g)} \mathbb{E}_{(x,y)}[\|x - y\|] \tag{4.14}$$

There exist sequences of distribution that do not converge under JS, KL, reverse KL or TV but do converge under EM distanced. It is shown in [21] that the Wasserestein distance defined in 4.14 is equivalent to $A + B$ term in Equation 4.7.

4.1.4 VGG19

The ImageNet project is a large visual database designed for use in visual object recognition software research. The ImageNet project runs an annual software contest, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), where software programs compete to correctly classify and detect objects and scenes [32].

AlexNet came out in 2012 and was a revolutionary advancement; it improved on traditional Convolutional Neural Networks (CNNs) and became one of the best models for image classification until VGG came out. While previous derivatives of AlexNet focused on smaller window sizes and strides in the first convolutional layer, VGG addresses another very important aspect of CNNs which is depth. VGG19 is an innovative object-recognition model that supports up to 19 layers. Built as a deep CNN, VGG19 also outperforms baselines on many tasks and datasets outside of ImageNet. VGG19 is still one of the most used image-recognition architectures [6]. We use VGG to estimate perceptual loss between the NDCT images generated by the generator network and the ground truth NDCT images.

	LDCT	GAN	WGAN [51]	Our Proposed WGAN method
PSNR	34.3094	37.2451	39.0300	38.2227
SSIM	0.8276	0.8967	0.9251	0.9126

TABLE 4.1: PSNR and SSIM of LDCT, GAN, WGAN [51] and our proposed WGAN method

4.2 Result

In this section we show results of the tests on our WGAN network. Similar to what we discussed in Chapter 3, we are using 10-time cross validation and Table 4.1 is the mean performance evaluation of these 10 time of training and testing the network.

Using VGG transfers knowledge of human perception that is embeded in VGG network to CT image quality evaluation [51]. The performance of using GAN alone is not acceptable, since it only maps the data distribution from LDCT to NDCT but does not guarantee the image content correspondence.

Figure 4.2, and 4.3 shows generator and discriminator loss. As we can see in Figure 4.3 we do not really know when to stop the training as there is no proper evaluation metric in GAN's training, and it is a drawback to this method. However, after 2000 iteration we observed that the generator is just fluctuating around a number so close to zero, therefore, we used 2000 ieteration as our training's end point. The plot of generator loss shows that the model has converged and has reasonable loss on this data-set.

For an example of one of the pieces of training with L298 as our validation file in Figure 4.4 a slice of abdominal CT in quarter-dose, NDCT, and Result/final output of our proposed WGAN network is presented from top to bottom, respectively. We can visually observe that the result of our proposed WGAN network is less fuzzy than the quarter dose. Moreover, we cannot perceptually tell the difference between NDCT and results of our network. Our proposed WGAN recovers more fine subtle details and captures more anatomical information.

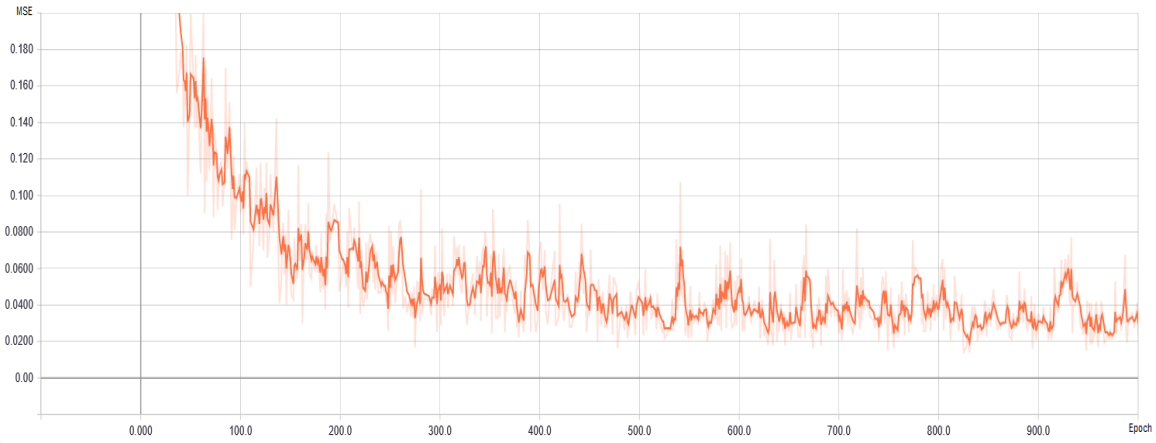


FIGURE 4.2: Loss function of generator shown for 1000 iterations

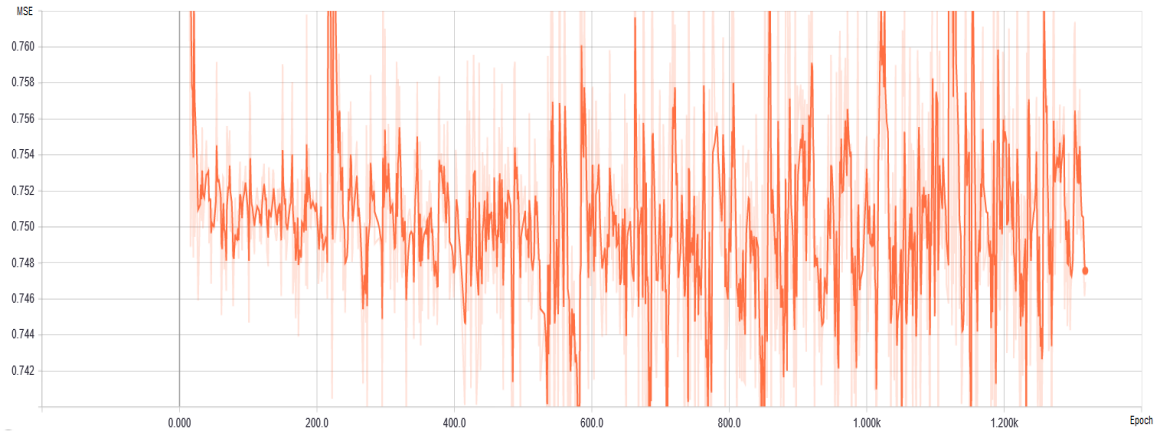


FIGURE 4.3: Loss function of discriminator shown for 1000 iteration

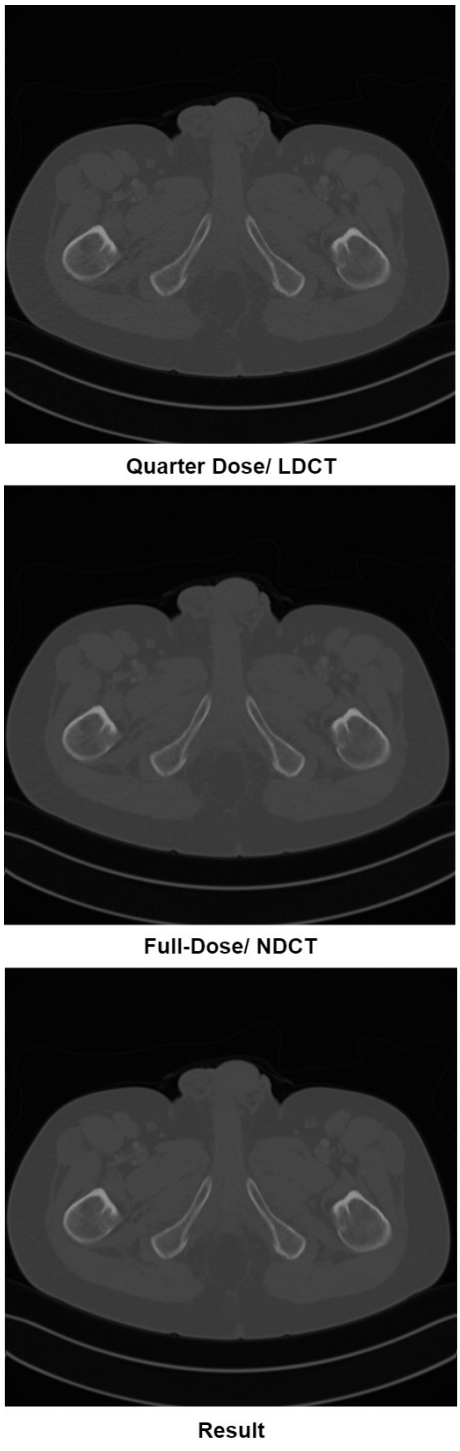


FIGURE 4.4: From top to bottom, Quarter-Dose/LDCT, Full Dose/NDCT and Result which is output of proposed residual network; All three are from a same slice of L298 patient's abdominal CT

Chapter 5

RES-GAN approach

Earlier in Chapter 3 and, Chapter 4, we discussed two recent state-of-art learning-based methods on LDCT image denoising: Residual learning and Generative Adversarial Network. Inspired by the works of [19] and [50], we propose a new method of low-dose CT denoising. GAN has traditionally demonstrated superior performance in terms of correlation with the human visual system. Although it has not yield better results in terms of quality measurements such as PSNR and SSIM, no one can deny visual superiority of GAN networks. On the other hand, Residual learning has shown an outstanding outcome in terms of quality metrics. Therefore combining these two networks, is capable of producing CT images numerically and visually similar to NDCT.

In a typical CNN, multiple downsampling can cause loss of valuable information in the image while passing through the network layers. To avoid this loss of image details, we used a residual CNN for our generator network. By combining GAN and residual CNN, we achieve outstanding results in LDCT denoising as will be presented later in the results section. The contributions of our proposed network are listed as follow:

- In general we are using a network consist of generator and discriminator, but for testing phase we put discriminator aside and use our generator alone to produce

images akin to normal dose, similar to what we have done for WGAN network in chapter 4.

- We use very deep residual network architecture consist of symmetric convolutional and deconvolutional layers for our generator.

- We add skip connections, with the same motivation in 3.2.4, between corresponding convolutional and deconvolutional layers.

Figure 5.1 shows the overall architecture of our approach for noise reduction of low-dose CT, called RES-GAN. As can be seen, our approach utilizes a generator and a critic box containing a pre-trained network VGG-19, and a discriminator network referred to as a Generative Adversarial network. Generally, Generative Adversarial Networks consist of two networks; Generator and Discriminator. Together they model the high-dimensional distribution of the data. The generator produces synthetic data and the discriminator, acting as an expert, tries to discover if the received data is either a sample of artificial or real data [19]. Here we are adding this VGG network as a feature extractor and using it as a new loss component, called VGG loss to our loss function.

5.1 Residual Generator

In Figure 5.1 in our proposed method, we use residual learning for our generative network. This network has ten layers, including five convolutional, and five deconvolutional layers. We removed pooling operation because denoising, unlike object recognition, is a low-level problem and does not need high-level feature extractor. Moreover, by using pooling layers image details that are essential for the denoising process can be eliminated [37]. We have 96 filters, size 3×3, and the last layer is size 1×1. Comparing results of using different sizes of filters, we empirically found 3×3 the most suited for

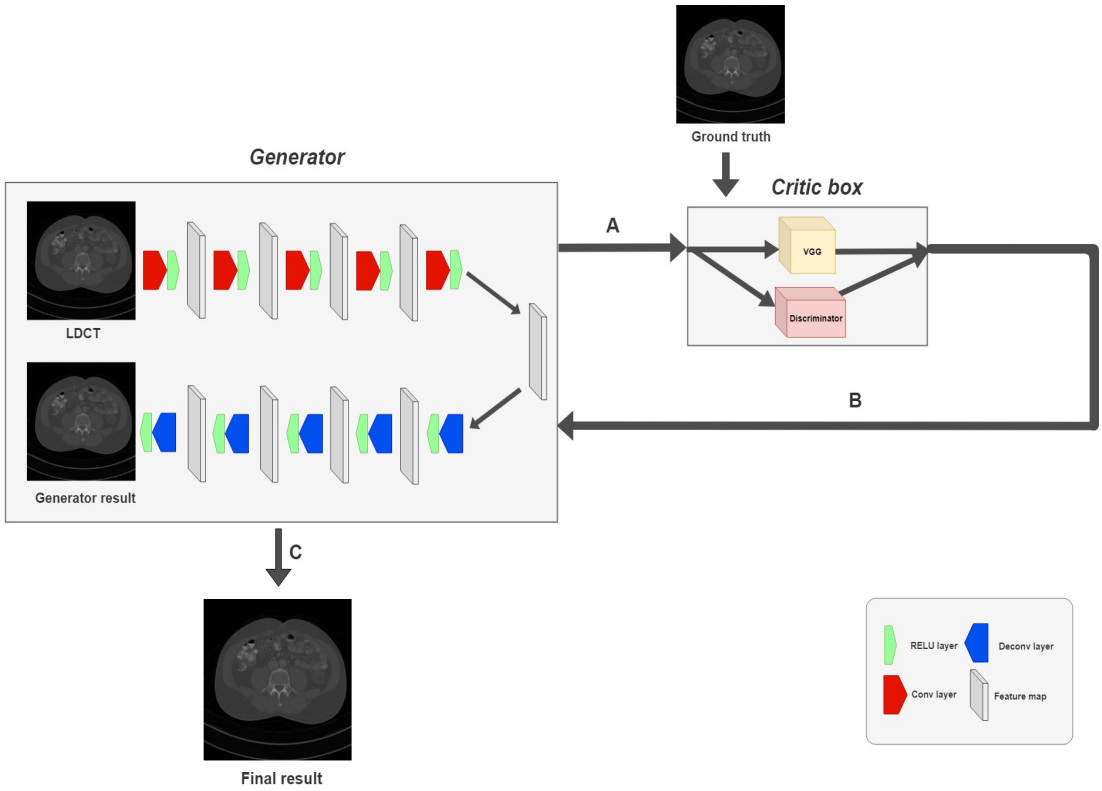


FIGURE 5.1: Overview of the proposed Architecture called RES-GAN for noise reduction of low-dose CT. The Generator consist of a RED-CNN with five convolutional and five deconvolutional layers using regression to determine the Normal-Dose and The critic box including discriminator and VGG-19 tries to characterize LDCT images from real NDCT

our generator network. To avoid boundary artifacts, we do not use any padding, which means the network is stride 1. Moreover, in Figure 5.1 A is the output image generated by our Res Generator. B is the discriminator feedback that we send to the generator to update its weights, and C is our final result for when we are testing the network, only using the generator and producing an NDCT image.

To avoid gradient vanishing problem, we also use skip connections between each convolutional layer with its corresponding deconvolutional one. We mentioned this earlier in Section 3.2.4. We set our patch size to 64×64, since it gave us better result in both residual network and GAN. We need to have the same kernel size for both encoder and

decoder [11], which in our case is 3×3 as we mentioned earlier to ensure the matching of network's input and output. After encoding the patches and extracting features, we need to recover image details by using deconvolutional layers. The error of the generator is computed using the generator, the discriminator, and the perceptual loss (using VGG19 pre-trained network). We are using Equation 4.7 to solve the minmax problem of this RES-GAN network

Every time that generator receives a feedback error, it changes the cost function and updates the values of each layer until the error is less than our set point. When the training job is complete, the work of discriminator is done too. At this time, we divide networks and use the only generator for producing NDCT.

5.2 Critic Box

Critic box consists of a discriminator and a VGG pre-trained network. The discriminator gets both NDCT and LDCT as input and determines whether it is real or not, meaning the probability of it being real. Convolutional layers organized in three blocks and each block contains two layers without stride and one layer with a stride of 2. We use LReLUs for activation function and batch normalization. Loss component for discriminator is an adversarial goal to differentiate LDCT and NDCT correctly.

The whole point of having a discriminator is that when we produce two images; one with use of a discriminator and the other with only a CNN-based method; both having similar PSNR and SSIM. The former is visually better than the latter. The reason behind it is that we yet do not have quality measurements comparable to human vision. After encoding the patches and extracting features, we need to recover image details by using deconvloutional layers. After we are done using discriminator for training the generator, we do not need the discriminator anymore.

Training environment	**Specification**
Patch size	64×64
Initializer	Random Gaussian distribution (0, 0.01)
Learning rate	10^{-6}
Number of iteration	2000
Optimizer	Adam
Loss function of G	MSE
Loss function of D	EM distance

TABLE 5.1: Hyper-parameters for training RES-GAN

VGG-19 is a pre-trained network that we are using for feature extraction and compare denoised output against the ground truth in terms of extracted features. We discussed VGG-19 with detail in Section 4.1.4.

5.3 Result

In this step of our work, we empirically use the best hyperparameters that we gained from testing both networks of Chapter 3 and Chapter 4, and utilize them to train our proposed RES-GAN network. Table 5.1 shows the details of our network training set.

To evaluate our proposed network we trained and tested our network ten times with cross-validation, changing the training and testing data-set 10 times, same in Chapter 3 and compute the mean PSNR and SSIM of these 10 times. In Table 5.2, we can see our proposed method along with methods from Chapter 3 and 4. As evident from the Table 5.2 our method has achieved the best result in terms of quantitative measurements.

As an example to show the training phase, we tested our network with another patient file that has not been used for training and results are shown in Figure 4.3 and Figure 4.2. In Chapter 4 we argued why discriminator is fluctuating and why we are not able to use it as our stopping point. However, as we can see in the Figure 4.2, the generator is converged well, and the loss function approaches zero.

Method	PSNR	SSIM
LDCT	34.3094	0.8276
BM3D	38.9903	0.9295
Our proposed Residual learning	40.0227	0.9473
Our proposed WGAN	38.2227	0.9126
RES-GAN	40.2378	0.9518

TABLE 5.2: PSNR and SSIM of different methods

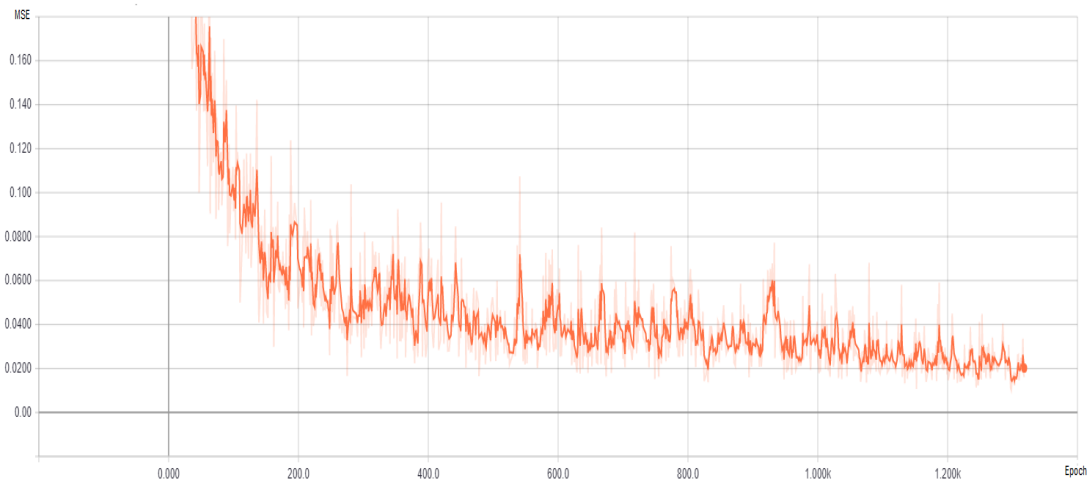


FIGURE 5.2: Generator loss function over training

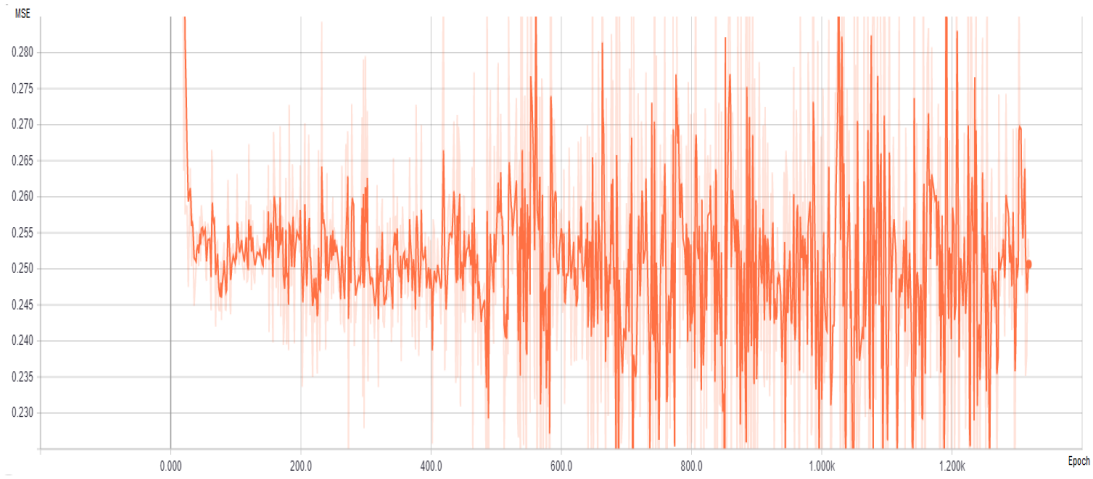


FIGURE 5.3: Discriminator loss function over training

Our data-set for both training and testing our network is the same *"NIH-AAPM-Mayo Clinic Low Dose CT Grand Challenge"* consists of contrast-enhanced abdominal CT examination, taken from 10 patients and the Table 3.1 shows all the patients' abdominal CT details.

In Figure 5.4 we demonstrate different slices of the abdominal CT from patient L067. In the left column are LDCT slices and infront of each, is the result of using proposed RES-GAN network to produce an NDCT image. It is shown that our method have a great potential of deep learning for noise suppression, structural preservation, and lesion detection. The networks using RES-GAN give better overall image quality.

As far as we are concerned, Residual CNN has better results in terms of PSNR, and SSIM compared to methods using GAN. However, GAN methods tend to have better perceptual results. Our proposed method is utilizing both Residual learning and WGAN, divine this problem. We achieve better performance in both quantitative and perceptual measurements. A significant advantage of the proposed method is its processing speed during the test since the discriminator CNN is only used during the training. Also, using residual learning helps speeding up the CNN too.

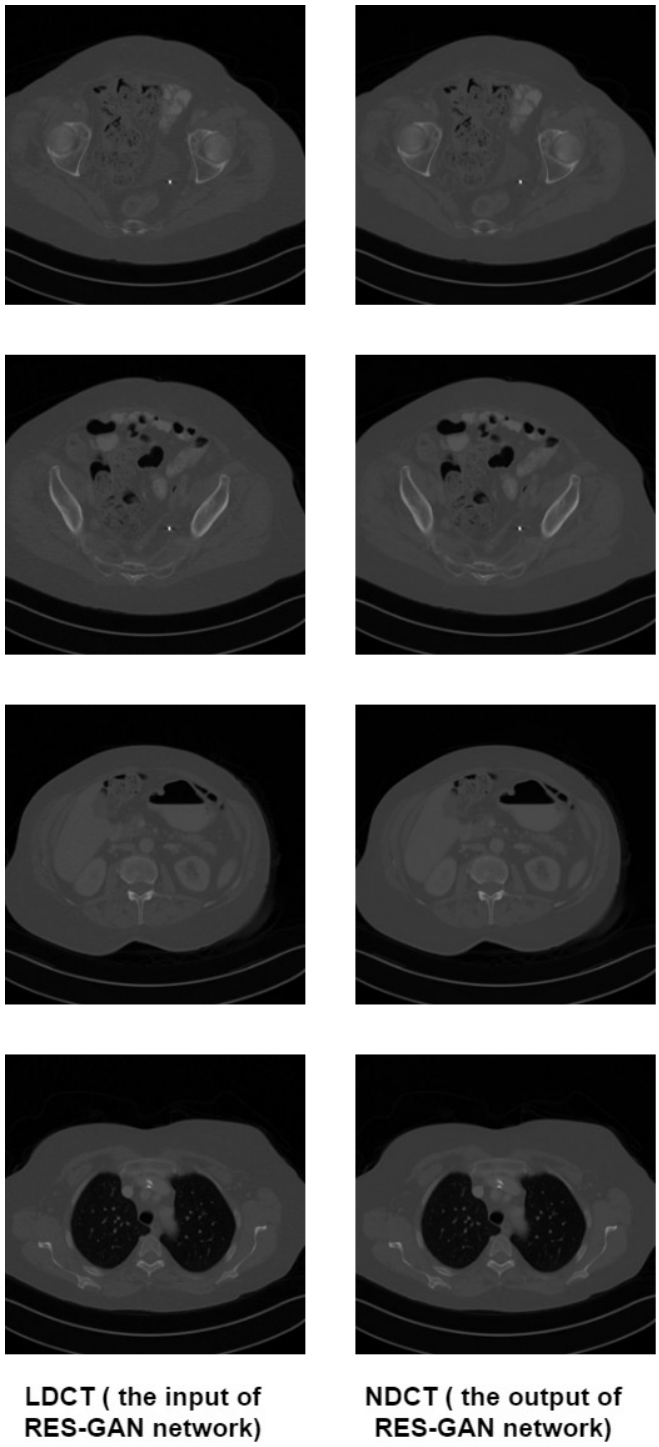


FIGURE 5.4: Results from abdominal CT for comparison, on the left there are four slices of different parts of abdominal LDCT and on the right are each of their outputs using RES-GAN

Chapter 6

Conclusion and Future work

Computed tomography (CT) is widely used for both clinical and research objectives. CT scan is a series of X-ray images which are taken from multiple angles of the tissue of the interest. X-ray beams, which is electromagnetic radiation with high intensity, can cause damage to cells, increasing the chance of complications such as cancer. Each X-ray procedure has a different level of risk associated depending on the type and amount of radiation as well as the body part being exposed. CT is identified as the most hazardous procedure among X-rays, and more specifically, abdominal CT puts the body in the most exposed state of radiation.

A possible solution proposed to tackle this problem is lowering the dosage of radiation. In low dose CT (LDCT) images, fundamental structures are still easily identifiable. However, noise and other artifacts are introduced. Removing the visual effects of artifacts caused by lowering radiation dose has been an active area of research in the last few years. The developed methodologies for denoising LDCT can be divided into three main classes: sinogram filtering, iterative reconstruction, and image processing.

A considerable disadvantage to sinogram filtering and iterative reconstruction methods is the scarcity of raw data. The advantage of post-processing methods over the other

two is the ability to perform without having access to raw data. In this thesis, we are working on post-processing method, which is working directly with CT images.

Majority of state-of-art approaches in the field of LDCT image denoising, deploy Convolutional Neural Networks (CNN), which have demonstrated promising performance as a result of their capability of learning noise characteristics. In our work, we examined three ML main approaches towards denoising abdominal LDCT, including residual deep learning, generative adversarial network, and RES-GAN, which is our proposed method and a combination of the first two.

Residual deep learning is a recent famous method for image denoising, which addresses the performance degradation problem caused by increasing the network depth. The objective in residual learning approach is to learn the noise (residual) maps rather than the actual NDCT images. The residual mapping is much easier to be learned than the original mapping, especially for networks that are substantially deeper than others. In this network we are using a combination of the autoencoder, and deconvolutional network.

The autoencoder was first introduced for unsupervised learning, but it is also useful for image denoising. Stacking multiple dense convolution layers, does not necessarily increase the learning rate. Techniques such as employing more suited activation functions (such as ReLU) might help. Nevertheless, the problem may still remain. Therefore we use skip connections which involves jumping over a number of layers. This technique allows us to increase the number of layers without hesitation.

Lack of sufficient labelled training data has always been a problem in the medical image domain. In field of medical image restoration, patch coding is commonly used in order to increase the number of traininig data. In our work, we extract fixed-size samples which correspond to patches from LDCT images and their corresponding NDCT

counterparts.

Our second approach is Generative Adversarial Network (GAN), which is starting to be a method of interest. One of the outstanding merits of GANs is that they generate data that is similar to real data, which results in having multiple applications in the real world. GANs can learn messy and complicated distributions of data, which can be helpful for many machine learning problems. The main disadvantages to GANs are, being hard to train and time-consuming. Loss component for our discriminator is an adversarial goal to differentiate LDCT and NDCT correctly. Our Proposed method is a combination of both Residual learning and GAN called RES-GAN. The advantage of our method is that we utilize high performance in both quantitative and perceptual measurements. A significant advantage of the proposed method is its processing speed during the test since the discriminator CNN is only used during the training. Also, using residual learning helps speeding up the CNN too.

In the learning of the mapping from LDCT to NDCT needs to estimate the weights of the convolutional and deconvolutional filters are determined. This is achieved by minimizing the cost function. We randomly extracted pairs of image patches from all of our CT training data-set images except the one we wanted to test with. We used cross-validation. Tests demonstrate that our RES-GAN outperforms other LDCT denoising methods in terms of PSNR and, SSIM.

6.1 Future Work

Our future work is to train different networks with CT image of different parts of the human body and determine which one works better for a particular section of the body. Then, by performing a classification before denoising we can get the best overall performance. Recently there has been a new data-set released for a new challenge, called

cheXpert. CheXpert is a large dataset of chest X-rays and competition for automated chest x-ray interpretation, which features uncertainty labels and radiologist-labeled reference standard evaluation sets. Our possible future is to work with this new data-set and try to increase our network performance.

Bibliography

[1] Forest Agostinelli et al. Learning activation functions to improve deep neural networks. *arXiv preprint arXiv:1412.6830* (2014).

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv preprint arXiv:1701.07875* (2017).

[3] David J Brenner and Eric J Hall. Computed tomography—an increasing source of radiation exposure. *New England Journal of Medicine* 357(22) (2007), 2277–2284.

[4] Antoni Buades, Bartomeu Coll, and Jean Michel Morel. On image denoising methods. *CMLA Preprint* 5 (2004).

[5] Harold C Burger, Christian J Schuler, and Stefan Harmeling. Image denoising: Can plain neural networks compete with BM3D. In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, 2392–2399.

[6] Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678* (2016).

[7] Baiyu Chen et al. An open library of CT patient projection data. In: *Medical Imaging 2016: Physics of Medical Imaging*. Vol. 9783. International Society for Optics and Photonics. 2016, 97831B.

[8] Hu Chen et al. Low-dose CT denoising with convolutional neural network. In: *Biomedical Imaging (ISBI 2017), 2017 IEEE 14th International Symposium on*. IEEE. 2017, 143–146.

[9] Hu Chen et al. Low-dose CT via convolutional neural network. *Biomedical optics express* 8(2) (2017), 679–694.

[10] Hu Chen et al. Low-dose CT with a residual encoder-decoder convolutional neural network. *IEEE transactions on medical imaging* 36(12) (2017), 2524–2535.

[11] Hu Chen et al. Low-dose CT with a residual encoder-decoder convolutional neural network. *IEEE transactions on medical imaging* 36(12) (2017), 2524–2535.

[12] Kostadin Dabov et al. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on image processing* 16(8) (2007), 2080–2095.

[13] Stanley R Deans. *The Radon transform and some of its applications*. Courier Corporation, 2007.

[14] Stanley R Deans. *The Radon transform and some of its applications*. Courier Corporation, 2007.

[15] Weisheng Dong et al. Nonlocally centralized sparse representation for image restoration. *IEEE transactions on Image Processing* 22(4) (2012), 1620–1630.

[16] Lucas L Geyer et al. State of the art: iterative CT reconstruction techniques. *Radiology* 276(2) (2015), 339–357.

[17] Amy Berrington de Gonzalez and Sarah Darby. Risk of cancer from diagnostic X-rays: estimates for the UK and 14 other countries. *The lancet* 363(9406) (2004), 345–351.

[18] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[19] Ian Goodfellow et al. Generative adversarial nets. In: *Advances in neural information processing systems*. 2014, 2672–2680.

[20] Shuhang Gu et al. Weighted nuclear norm minimization with application to image denoising. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, 2862–2869.

[21] Ishaan Gulrajani et al. Improved training of Wasserstein GANs. In: *Advances in neural information processing systems*. 2017, 5767–5777.

[22] Mohammad Mahedi Hasan. Adaptive Edge-guided Block-matching and 3D filtering (BM3D) Image Denoising Algorithm (2014).

[23] Simon S Haykin et al. *Neural networks and learning machines/Simon Haykin*. New York: Prentice Hall, 2009.

[24] Kaiming He et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, 770–778.

[25] Alain Hore and Djemel Ziou. Image quality metrics: PSNR vs. SSIM. In: *2010 20th International Conference on Pattern Recognition*. IEEE. 2010, 2366–2369.

[26] Gareth James et al. *An introduction to statistical learning*. Vol. 112. Springer, 2013.

[27] Mannudeep K Kalra et al. Low-dose CT of the abdomen: evaluation of image improvement with use of noise reduction filters—pilot study. *Radiology* 228(1) (2003), 251–256.

[28] Eunhee Kang, Junhong Min, and Jong Chul Ye. A deep convolutional neural network using directional wavelets for low-dose X-ray CT reconstruction. *Medical physics* 44(10) (2017), e360–e375.

[29] Eunhee Kang, Jong Chul Ye, et al. Wavelet domain residual network (WavResNet) for low-dose X-ray CT reconstruction. *arXiv preprint arXiv:1703.01383* (2017).

[30] Bekir Karlik and A Vehbi Olgac. Performance analysis of various activation functions in generalized MLP architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems* 1(4) (2011), 111–122.

[31] Andrej Karpathy. *Neural Networks*. 2019. URL: https://karpathy.github.io/ (visited on 07/11/2019).

[32] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. 2012, 1097–1105.

[33] Gaurav Kumar and Pradeep Kumar Bhatia. A detailed review of feature extraction in image processing systems. In: *2014 Fourth international conference on advanced computing & communication technologies*. IEEE. 2014, 5–12.

[34] Geert Litjens et al. A survey on deep learning in medical image analysis. *Medical image analysis* 42 (2017), 60–88.

[35] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In: *Proc. icml*. Vol. 30. 1. 2013, 3.

[36] Julien Mairal et al. Non-local sparse models for image restoration. In: *ICCV*. Vol. 29. Citeseer. 2009, 54–62.

[37] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In: *Advances in neural information processing systems*. 2016, 2802–2810.

[38] Cynthia H McCollough et al. Strategies for reducing radiation dose in CT. *Radiologic Clinics* 47(1) (2009), 27–40.

[39] Jawad Nagi et al. Max-pooling convolutional neural networks for vision-based hand gesture recognition. In: *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*. IEEE. 2011, 342–347.

[40] David P Naidich et al. Low-dose CT of the lungs: preliminary observations. *Radiology* 175(3) (1990), 729–731.

[41] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010, 807–814.

[42] NIBIB. *Computed Tomography*. 2000. URL: https://www.nibib.nih.gov/science-education/science-topics/computed-tomography-ct (visited on 06/26/2019).

[43] Rajni Rajni and Anutam Anutam. Image denoising techniques-an overview. *International Journal of Computer Applications* 86(16) (2014), 13–17.

[44] Hariharan Ravishankar et al. Understanding the mechanisms of deep transfer learning for medical images. In: *Deep Learning and Data Labeling for Medical Applications*. Springer, 2016, 188–196.

[45] Jing Wang et al. Sinogram noise reduction for low-dose CT by statistics-based nonlinear filters. In: *Medical Imaging 2005: Image Processing*. Vol. 5747. International Society for Optics and Photonics. 2005, 2058–2067.

[46] Yifan Wang et al. End-to-end image super-resolution via deep and shallow convolutional networks. *IEEE Access* 7 (2019), 31959–31970.

[47] Zhou Wang et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13(4) (2004), 600–612.

[48] Warren Weaver. Recent contributions to the mathematical theory of communication. *ETC: a review of general semantics* (1953), 261–281.

[49] *what is voxel Voxel definition*. https://whatis.techtarget.com/definition/voxel. Accessed: 2019-07-18.

[50] Jelmer M Wolterink et al. Generative adversarial networks for noise reduction in low-dose CT. *IEEE transactions on medical imaging* 36(12) (2017), 2536–2545.

[51] Qingsong Yang et al. Low-dose CT image denoising using a generative adversarial network with Wasserstein distance and perceptual loss. *IEEE transactions on medical imaging* 37(6) (2018), 1348–1357.

[52] Kai Zhang et al. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing* 26(7) (2017), 3142–3155.

[53] Kai Zhang et al. Learning deep CNN denoiser prior for image restoration. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2017, 3929–3938.

[54] Yi Zhang et al. Statistical iterative reconstruction using adaptive fractional order regularization. *Biomedical optics express* 7(3) (2016), 1015–1029.