# Adopting Cross Channel Communication Block in to the Generative Adversarial Network for Denoising of Low Dose CT Images

Tihetna Mesfin Gebeyehu

A Thesis Submitted to the Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

September 2022

Adama, Ethiopia

# Adopting Cross Channel Communication Block in to the Generative Adversarial Network for Denoising of Low Dose CT Images

Tihetna Mesfin Gebeyehu

Advisor: Worku Jifara Sori (Ph.D)

A Thesis Submitted to the Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

September 2022

Adama, Ethiopia

# Declaration

I hereby declare that this Master Thesis entitled "**Adopting Cross Channel Communication Block in to the Generative Adversarial Network for Denoising of Low Dose CT Images"** is my original work. That is, it has not been submitted for the award of any academic degree, diploma, or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation.

Tihetna Mesfin _____ _____

Name Signature Date

# Recommendation

I, the advisor of this thesis, hereby certify that I have read the revised version of the thesis entitled "**Adopting Cross Channel Communication Block in to the Generative Adversarial Network for Denoising of Low Dose CT Images**" prepared under my guidance by Tihetna Mesfin submitted in partial fulfillment of the requirements for the degree of Mater's of Science in Computer Science and Engineering. Therefore, I recommend the submission of a revised version of the thesis to the department following the applicable procedures.

Dr. Worku Jifara (Ph.D) _____ _____

Major Advisor Signature Date

# Approval Page

I, the advisor of the thesis entitled "**Adopting Cross Channel Communication Block in to the Generative Adversarial Network for Denoising of Low Dose CT Images**" and developed by **Tihetna Mesfin Gebeyehu**, hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Dr. Worku Jifara (Ph.D) _____ _____

Major Supervisor Signature Date

We, the undersigned, members of the Board of Examiners of the thesis by **Tihetna Mesfin Gebeyehu** have read and evaluated the thesis entitled "**Adopting Cross Channel Communication Block in to the Generative Adversarial Network for Denoising of Low Dose CT Images**" and examined the candidate during the open defense. This is, therefore, to certify that the thesis is accepted for the partial fulfillment of the requirement of the degree of Master of Science in Computer Science and Engineering.

| | | |
|---|---|---|
| Chairperson | Signature | Date |
| Internal Examiner | Signature | Date |
| External Examiner | Signature | Date |

Finally, approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

| | | |
|---|---|---|
| Department Head | Signature | Date |
| School Dean | Signature | Date |
| Office of Graduate Studies, Dean | Signature | Date |

# ACKNOWLEDGMENT

Above all, I want to praise and thank the **HOLY TRINITY FATHER**, **SON**, and **HOLY SPIRIT** the Almighty **GOD** who created everything visible and invisible. He has given me strength and encouragement throughout all the challenging moments of completing this thesis work. Also, I love to praise the Virgin St. Mary, the Holy Mother of **LORD JESUS CHRIST**, and all angels and saints as I couldn't finish my thesis work without their prayers and support for me.

How does a person say thank you when there are so many people to thank? However, some should be honored for the crucial support they provided during the course of study.

I especially like to thank my thesis advisor, Dr. Worku Jifara (Ph.D.), for his constant advice, encouragement, and suggestions throughout the entire thesis work. I want to express my gratitude to Prof. Yun Koo Chung (Ph.D.), head of the Computer Vision and Robotics SIG, for his patient advice on any matter and the suggestions he has given since the start of this research.

My deepest thanks to Duguma Yeshitila for his steadfast assistance in providing me with resources. Sincere appreciation to Anteneh Tilaye (M.Sc.) for his insightful comments, sage guidance, and generous support. I am also grateful for all the support and advice I have received from senior computer vision postgraduate members, and friends.

Lastly, my profound gratitude also extends to my entire family, including my mother Mulatua Arega, my father Mesfin Gebeyehu, all of my siblings, and close companions for their support and counsel.

# Table of Contents

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF CODE SNIPPETS

# LIST OF ABBRIVATION AND ACRONYMS

| | |
|---|---|
| AdaIn | Adaptive Instance Normalization |
| AI | Artificial Intelligent |
| ALARA | As Low As Reasonably Attainable |
| CA | Channel Attention |
| CTA | Cardiac multiphase coronary Computerized tomography |
| CNN | Convolutional Neural Network |
| CT | Computerized tomography |
| DNA | Deoxyribonucleic acid |
| DU-GAN | Dual Domain Unet Generative Adversarial Network |
| DRWGAN-PF | Dilated convolution, Residual structure, Wasserstein General Adversarial Network - Perceptual loss and Fidelity loss |
| ECA | Efficient Channel Attention |
| EMI | Electromagnetic Interference Laboratory |
| FBP | Filtered Back-Projection |
| GAN | Generative Adversarial Network |
| GAP | General Adaptive Pooling |
| GPU | Graphical processing unit |
| IN | Instance Normalization |
| LDCT | Low dose Computerized tomography |

| | |
|---|---|
| LS-GAN | Least Squared Generative Adversarial Network |
| L1 | Manhattan Distance or L1 Norm |
| MPGD | Mixed Poisson Gaussian distribution |
| MSE | Mean Squared Error |
| mSv | Millisieverts |
| NBIA | National Biomedical Imaging Archive |
| NDCT | Normal dose Computerized tomography |
| NIHAAPM | National Institutes of Health American Association of Physicists in Medicine |
| PCM | Pulse-code modulation |
| PSNR | Peak Signal to Noise Ratio |
| RAM | Random Access Memory |
| REDCNN | Residual Encoder Decoder Convolutional Neural Network |
| ReLU | Rectified Linear Unit |
| ResNet | Residual Blocks |
| RMSE | Root Mean Squared Error |
| ROI | Region Of Interest |
| SA-CNN | Self Attention - Convolutional Neural Network |
| SAGAN | Self Attention - Generative Adversarial Network |
| SCN | Stacked Competitive Network |
| SENet | Squeeze and Excitation Network |
| SSIM | Structural Similarity Index Measure |

| | |
|---|---|
| WGAN | Wasserstein Generative Adversarial Network |
| WGAN-VGG | Visual Geometry Group |
| WHO | World Health Organization |
| VGG | Visual Geometry Group |
| 2D | 2-Dimension |
| 3D | 3-Dimension |

# LIST OF NOTATIONS AND SYMBOLS

| Notation | Meaning |
|---|---|
| $T_{ld}$ | Low-dose transmission data |
| $T_{nd}$ | Normal-dose transmission data |
| $\mu_{water}$ | Water attenuation coefficients |
| $\mu_{nd}$ | Linear attenuation coefficients |
| $\rho_{ld}$ | Low-dose projection data |
| $\rho_{nd}$ | Normal-dose projection data |
| $\rho_{noise}$ | Projection of the added Poisson noise |
| H $\cup_{nd}$ | Hounsfield unit numbers of the normal dose CT image |
| Iradon | Inverse Radon |
| $I^{\circ}$ | Incident flux |
| $I_{ld}^{\circ}$ | Simulated low-dose scan incident flux |
| $D_{dec}$ | Discriminator decoder |
| $D_{enc}$ | Discriminator encoder |
| $\lambda_{adv\_loss}$ | Loss weight for adversarial loss |
| $\lambda_{img}$ | Loss weight for dice loss in image domain discriminator |
| $\lambda_{grd}$ | Loss weight for L1 in gradient domain discriminator |
| et.al | and others |
| $\nabla$ | Sobel operator |

# ABSTRACT

*The CT's continued advancement and wide-ranging application in medical operations have prompted questions over the radiation dose connected to patients. Since low-dose CT (LDCT) may reduce the radiation risk, it has drawn a lot of attention. However, merely reducing radiation dosages will cause a conspicuous decline in image quality. Consequently, reducing the noise in the LDCT image is a preferable solution to this problem. Numerous studies have used GANs to significantly improve LDCT image denoising. Current GAN-based LDCT image denoising approaches struggle to provide better structural detail retention. This is mostly due to model training not placing enough emphasis on pixel-wise loss function tolerance and structural feature retention. This study investigates the use of cross-channel communication via an adequate attention mechanism, i.e. efficient channel attention, by extending the recently developed deep learning model known as DU-GAN with dual domain U-Net discriminators, which outperforms the leading models in the field of LDCT restoration. Moreover, a lot of work was put forward to combine the Dice loss function which guides the denoising process in the image domain discriminator, to ensure that the final denoised results are as close as feasible to the industry's gold standard, NDCT. The aforementioned extensions are combined with instance normalization to increase training effectiveness by minimizing the statistical difference between the CT images. The results of our trials using actual clinical images reveal that the proposed method has improved when compared to other experiments and measurements. The suggested model improves the DU-GAN baseline work from 0.72109 to 0.73830, from 20.40010 to 22.04259, and from 0.09821 to 0.08170, respectively, based on SSIM, PSNR, and RMSE. This study finds that integrating effective channel attention into the generator network and utilizing Dice loss as a pixel-wise loss in image domain discriminator instead of Mean Absolute Error loss increase the quality of the denoised LDCT images.*

***Keywords**: DU-GAN, LDCT, NDCT, U-Net, Denoising, Dice loss, over-smoothing, structural detail preservation*

# CHAPTER ONE

# 1. INTRODUCTION

## 1.1. Background of the study

Godfrey Hounsfield, a British engineer of EMI Laboratories, created the first CT scanner that was readily accessible for purchase in 1972. He co-developed the technology with physicist Dr. Allan Cormack.[1] Ever since the CT scanner was applied in medical diagnosis, a lot of advancements and developments in technology are being used. Recent advancements in CT scan technology include decreasing the harmful radiation that patients are exposed to, enabling quicker scan speeds, and improving image quality. These advancements are made possible by the use of various computing techniques and computer vision.

The importance of computer vision in medicine is growing at an exponential rate nowadays. Computer vision has been utilized in a variety of healthcare applications to help medical practitioners make better judgments about patient treatment and reduce the dangers associated with diagnostic equipment such as CT scanners(Gerig, Kuoni, Kikinis, & Kübler, 1989). One of the areas of research that continue to attract attention in the medical imaging field is low-dose CT denoising.

According to the world health organization (WHO) report, medical usage of radiation accounted for 98 percent of the population contribution from all artificial sources, and 20 percent of overall population exposure. Over 3600 million diagnostic radiological examinations, 37 million nuclear medicine operations, and each year 7.5 million radiation treatments are carried out globally.[2] The extent of radiation damage to tissue and/or organs is determined by the dose of radiation received. Radiation can impair the functioning of tissues and/or organs above certain thresholds, resulting in the development of solid organ malignancies, leukemia, and other cancer diseases. These side effects are more severe at higher doses and higher dose rates. To conquer this, the best alternative is low dose CT (LDCT), which uses only 1.4 millisieverts (mSv) of radiation compared to a standard diagnostic CT scan's 7 mSv for the same test.[3]



1 https://catalinaimaging.com/history-ct scan/

2https://www.who.int/news-room/fact-sheets/detail/ionizing-radiation-health-effects-and-protective-measures

3https://www.cancercenter.com/cancer-types/lung-cancer/diagnosis-and-detection/low-dose-ct-scan

Even though LDCT is a better option for reducing the radiation risk, it highly affects the image quality which results in jeopardizing the clinical diagnostic performance as well as automated analysis tasks such as segmentation, feature extraction, and classification of these LDCT images have suffered. (Kaur, Juneja, & Mandal, 2018) Reducing the noise in the LDCT image (I. Goodfellow et al., 2020) is a preferable alternative to address this problem. Due to its ill-posed nature, however, it remains a difficult problem.

Various denoising algorithms have been presented during the last 5 decades. Sinogram domain filtering, iterative reconstruction, and picture domain processing (Mohd Sagheer & George, 2020) are some of the algorithms that fall within this category. In general, these approaches convert LDCT images into Normal-Dose CT (NDCT) representations. The LDCT restoration is hindered, nevertheless, by the scarcity of projection data in the sinogram domain and the high cost of computing in the iterative reconstruction domain. Image domain processing, on the other hand, uses an image post-processing technique and does not rely on projection data, making it easy to test new methodologies for improving denoising quality.

Earlier methods for denoising Low dose CT images using image post-processing techniques with traditional approaches, such as non-local means (Yuan, Zhang, & Yu, 2018), (Yanbo Zhang, Salehjahromi, & Yu, 2019), total variation (Jia et al., 2018), Block Matching Three Dimension (BM3D) (Hashemi, Paul, Beheshti, & Cobbold, 2015), and statistics-based algorithms (Geraldo, Cura, Cruvinel, & Mascarenhas, 2016), the majority of them are unable to compute noise statistics due to non-uniformity, and the proposed CT restoration applications will be inaccurate. Furthermore, structural information in CT scans is largely obscured. As a result, these LDCT restoration methods and their limitations have paved the way for novel deep learning-based LDCT restoration approaches to be developed.

To deal with LDCT denoising, recent techniques rely on generative models. GAN (generative adversarial network) (B. I. Goodfellow et al., 2014) is one of the most prominent, having made great progress in denoising LDCT images. Due to their multi-objective functions, ability to update generator architectures, and multi-scale discriminator, GAN-based applications outperform alternative deep Learning based LDCT restoration techniques, according to analysis results. However, when comparing the quantitative results of those offered approaches based on PSNR and SSIM measurements, the quantitative results of those proposed methods are not still

optimal in some circumstances. A significant issue that comes from model training is the disregard for structural feature preservation and tolerating pixel-wise loss functions (Kulathilake, Abdullah, Sabri, & Lai, 2021).

To alleviate this problem, inspired by the human visual perception (Balda, Hornegger, & Heismann, 2012) and attention model (Ouyang, Solberg, & Wang, 2011) for image recognition, some works (Du, Chen, Liao, Yang, & Wang, 2019) incorporate the visual attention mechanism into the generative adversarial network (GAN) framework to improve the performance of noise reduction and structural detail preservation. To obtain better performance, these studies use more sophisticated attention modules, i.e. channel attention, but this inevitably increases model complexity because dimensionality reduction eliminates the direct connection between a channel's weight and its channel (Q. Wang et al., 2020).

Moreover, this dimensionality reduction makes it harder to learn to channel attention and engage in proper cross-channel communication, which is one of the reasons these attention modules become increasingly sophisticated. Recent work (Huang, Zhang, Zhang, & Shan, 2022) uses a U-Net-based discriminator for LDCT denoising, which can simultaneously learn global and local differences between the denoised and normal-dose images, in contrast to existing GAN-based denoising approaches, and as a result, It gets superior performance over recently published methods both qualitatively and quantitatively. (Kulathilake et al., 2021) Kulathilake et al. claim that although the current attention-based DL methods have gained an acceptable visual performance in LDCT restoration, they are still under constraints for improving the performance of LDCT restoration. This thesis work proposed to extend Huang et al., work to improve structural detail preservation by adding extra cross-channel communication blocks via effective channel attention blocks since it avoids dimensionality reduction and effectively captures cross-channel interaction. It also proposed the introduction of dice loss as a pixel-wise loss function for better-denoised results.

## 1.2. Motivation for the Study

Health is undeniably important in life and ensuring the safety of humans is a critical duty, that is why extensive discoveries and technologies have been investigated in this area ever since the beginning of human history. Even though these advancements bring a lot of achievement in the

sector, some of them have undeniable side effects. Therefore, lowering dangers associated with diagnostic techniques that make use of advanced technology is another concern nowadays.

Due to the abundance of the vast quantity of data available nowadays, recent deep learning advancements becomes crucial for medical imaging. However, there is still a significant issue with the collection of medical data, particularly for research areas like denoising LDCT images, which requires paired datasets (such as LDCT and NDCT images) since creating an LDCT image requires methods based on the projection data that are usually unavailable in real applications. LDCT simulation as a transformation from a Normal dose CT (NDCT) image to the LDCT images using GAN could be the mechanism to overcome such problems. After the first paper on GAN by Goodfellow et al. (B. I. Goodfellow et al., 2014), GAN has been used in a wide range of areas for numerous applications including medical imaging and incorporating the visual attention mechanism into the generative adversarial network maybe is the significant one in this area.(Du et al., 2019) Advancement in GAN enables scientists and researchers to create fake images indistinguishable from real ones (Karras, Laine, & Aila, 2019)(Wang et al., 2018).

By extending the visual attention idea, various researchers propose a number of approaches for the LDCT denoising network to provide a powerful prior of the noise distribution. By doing this, both the generator and discriminator networks are empowered with visual attention information so that they will not only pay special attention to noisy regions and surrounding structures but also explicitly assess the local consistency of the recovered region. But most of the visual attention approaches failed (Kulathilake et al., 2021) to achieve the potential found by denoising networks because of the attention mechanisms used are more sophisticated which inevitably increase model complexity and they faced a problem of lack of structural detail preservation due to lack of local cross channel interaction strategy without dimensionality reduction. Besides the issues with attention mechanisms, lack of tolerating pixel-wise loss are still open for further investigation in the LDCT restoration problem. This work tries to solve these problems by extending the solutions presented in previous works.

As a result, it's incredibly motivating to be able to contribute in a solution making in the area of LDCT restoration, which becomes viable approach for reducing such diagnosis risk for patients while having no impact on doctors' diagnostic decision-making.

## 1.3. Statement of the Problem

CT scans are widely used in clinical, industrial, and other applications. Although some may claim that the dangers of commercial CT scans are exaggerated, the tremendous increase in CT use has already raised the global annual cumulative ionizing radiation dosage by 34% (Tahmasebzadeh, Paydar, Soltani-kermanshahi, & Reiazi, 2021). According to studies, this type of radiation can harm the DNA and cause cancer. The well-known guiding principle of ALARA (As Low As Reasonably Attainable) (Brenner & Hall, 2009) has aroused researchers' interest in CT dosage reduction. Lower the operating current of an X-ray tube and minimize the exposure time to reduce the X-ray flux. The lesser the X-ray flux, the messier the reconstructed CT image, lowering the signal-to-noise ratio and thus jeopardizing diagnostic performance.

Sinogram domain filtering, iterative reconstruction, and image domain restoration (Mohd Sagheer & George, 2020) are major categories of algorithms that have been developed to improve the image quality of low-dose CT scans (LDCT) to overcome this intrinsic physical difficulty. However in LDCT restoration, performance is a crucial criterion that keeps growing. In this sense, there are various knowledge gaps to fill in the existing restoration domain to improve the quality of LDCT images. Most previous works try to address these gaps by applying different approaches in different deep learning models and one is applying attention networks, which are a deep learning technique for enhancing the effectiveness of LDCT restoration and have lately gained popularity. However, the quantitative results of those proposed attention methods in these studies (Du et al., 2019),(M. Li, Hsu, Xie, Cong, & Gao, 2020) are not optimal and this is mainly because of the lack of attention given to the structural detail preservation and tolerate the pixel-wise loss functions during the model training.

These methods therefore employ more complicated attention modules, that is, channel attention, in order to achieve higher performance, but doing so inevitably results in a more complex model since dimensionality reduction weakens the connection between a channel's weight and its channel (Q. Wang et al., 2020). Thus, it becomes more challenging to participate in effective cross-channel communication, which is crucial to learn to focus attention to structural details.

Inspired by the work (Du et al., 2019) that suggests “to give a powerful prior of the noise distribution by incorporating visual attention knowledge into the learning method of GAN” this thesis proposes a method adopting state of the art attention mechanism that is, efficient channel

attention, that improves the structural detail retention by avoiding dimensionality reduction and providing enhanced local cross channel communication, in to recent work (Huang et al., 2022), that utilized U-Net-based discriminators in the GANs framework to learn both global and local differences between denoised and normal-dose images in both image and gradient domains. Besides this work proposes a pixel-wise loss that is, dice loss, which allows the denoising network to increase the specific features required by the discriminator network and helps the network to tolerate large noise kernels during model training, to see more enhancement in the quality of LDCT images both visually and in evaluation metrics. And to achieve this, an extensive experimental endeavor was made with the purpose of answering the research questions.

## 1.4. Research Question

The following research questions will be addressed by this study.

RQ1. What are the effective learning constraints to improve the quality of denoised LDCT images?

RQ2. Can adopting a cross-channel communication block improves the quality of denoised LDCT images?

RQ3. What effect would using dice loss as a pixel-wise loss have on the quality of denoised LDCT images?

## 1.5. Objective of the Study

### 1.5.1. General Objective

The general objective of this study is to improve the denoising quality of low dose CT scan (LDCT) images using a generative adversarial network (GAN).

## 1.5.2. Specific Objective

To achieve the general objective of the study; the following specific objectives have been established.

- To identify what effective learning constraint should be adopted to improve the quality of denoised LDCT images.
- To embed an attention mechanism, that can provide effective cross-channel communication, to the learning GAN model for achieving better reconstruction fidelity of low-dose CT datasets.

- To design GAN based model by adopting effective cross-channel interaction block via attention mechanisms, and adding instance normalization to enhance the denoising process.
- To embed dice loss to the learning GAN model.

## 1.6. Scope and Limitation of the Study

### 1.6.1. Scope of the Study

The following topics are covered within the thesis study's scope given the time and resources at hand:

- Adopting attention module in the generator network that effectively handles cross-channel communication to enhance the quality of denoised LDCT image.
- Embed dice loss as pixel-wise loss between the NDCT images and denoised LDCT images, in image domain discriminator branch.
- Adding instance normalization to improve training effectiveness by reducing the statistical difference between the CT images and for faster convergence and reduce sensitivity to initiate the learning model.

### 1.6.2. Limitation of the Study

Limitations of the Study includes information lost during the Filtered Back-Projection (FBP) reconstruction cannot be easily recovered because the suggested methodology is a post-processing method, which is a drawback of all post-processing techniques. Furthermore, this thesis work does not conduct experiments on several datasets with various amounts of noise due to resource and time limitations.

## 1.7. Significance of the Study

This work could be one of the contributions in the area of denoising LDCT images with some enhancements and full integration of the suggested model. So that the radiologist and/or physicians may get a better diagnosis and make better decisions while the patient's exposure to radiation due to the high dose CT scan is lowered. Furthermore, the work can be a valuable resource for scientists conducting additional research and investigation in this field.

## 1.8. Contribution of the Study

To the best of our knowledge this thesis study has a contribution to make in the area of LDCT restoration. The following is recorded as this work's contribution:

- Enhancing the quality of denoised LDCT image by adopting attention mechanism, that could provide effective cross channel communication block, in to the network to enrich the structural detail information while denoising is being processed.
- Reducing the blurriness and over smoothing traits, that are occurred as the denoising network attempts to eliminate the striking artifacts, by introducing the dice loss function as pixel wise loss in the image domain discriminator.
- Instance normalization has been introduced in the generator network to mitigate the statistical difference between the CT images problems and for faster convergence as well as for stable training of the learning model.

## 1.9. Organization of the Thesis

The rest of the thesis is structured as follows:

Chapter Two: The background literature and related works for Low Dose CT Denoising are discussed in this chapter. This chapter also elasticizes the Deep Learning and Generative Adversarial Network theoretical frameworks.

Chapter Three: includes the research methodology, which includes the various approaches and procedures needed to develop and choose the best answer. Methodologies of data collecting, data preprocessing, design and development frameworks and tools, and evaluation methods are all covered.

Chapter Four: will go over the suggested learning function in great depth. Model architecture, learning parameters, and pseudo code for training.

Chapter Five: Explains how the recommended solution is put into action. Regarding the creation of the working environment, proposed model implementation, and training implementation utilizing snippet codes of the proposed solution.

Chapter Six: The testing results from the proposed solution are explained and compared to other related work to arrive at the best conclusion.

Chapter Seven: The study comes to a close with recommendations for future research.

# CHAPTER TWO

# 2. LITERATURE REVIEW

## 2.1. Introduction

Back in the day, the program was hard-coded with instructions and rules represented by mathematical formulas. As they try to imitate complex real-world circumstances and data, such approaches fall short. To address this scarcity, a new software development paradigm arises, which examines if a computer can understand as well as a person by being cognitively trained called Artificial intelligence or AI (Vanderplas, n.d.).

Machine learning, at its most basic level, entails the creation of mathematical models to aid in the comprehension of data. When we provide these models with configurable parameters that may be altered to observed data, we can consider the program to be "learning" from the data. Machine learning may be divided into two forms at the most fundamental level: supervised and unsupervised learning. Supervised learning entails creating a model for the connection between measured data properties and a label associated with the data, which can then be used to apply labels to fresh, unknown data. Unsupervised learning, sometimes known as "letting the dataset speak for itself," is modeling the features of a dataset without any reference to a label. There are further approaches known as semi-supervised learning, which lie between supervised and unsupervised learning. When just partial labels are available, semi-supervised learning approaches are frequently beneficial (Vanderplas, n.d.). Algorithms for machine learning have been employed in a range of applications. Researchers have made several attempts to improve the accuracy of machine learning systems. Another factor was taken into account, resulting in the concept of deep learning (Shinde, 2018).

Deep learning is a branch of machine learning that concentrates on self-improvement via examining computer algorithms. Deep learning will undoubtedly be used to solve problems in a variety of new application domains and sub-domains. Deep learning approaches have recently gotten a lot of attention and have outperformed many classical algorithms in a variety of fields (Shinde et al., 2009),(I. Goodfellow, n.d.).The artificial neural network ANN was the first deep learning system to emerge. Bengio et al. (Bengio & Haffner, 1998) developed the Convolutional

Neural Network (CNN) in 1989 to detect handwritten digits was limited by a lack of dataset and poor processing capabilities at the time. CNN grew in popularity when AlexNet (Krizhevsky & Hinton, n.d.) won the Image Net competition in 2012 with the image classification network. The work of Alex et al. has opened the eyes of many scientists and academics to the capabilities of deep learning.

Denoising is one of the many applications where deep learning has been used. It has been a prominent research subject in the area of medical imaging as one of the inevitable degradation factors in this sector is noise and artifacts. In the context of medical imaging, applying the denoising methods which applied for natural image is not optimal. The main reason for this is, the medical images are usually represented as texture-rich low-contrast images than natural images and they rely heavily on training samples rather than noise type. To address this issue incredible various deep learning architectures are being constructed which outperform typical denoising filters. Deep learning algorithms have a lot of layers, and information flows through each one. Each layer can extract features and pass them on to the next. The first layers extract low features, and the next layers combine the features to form a complete representation. Thus, this high-level feature capturing of deep learning models demonstrates its ability to learn the uncertain noise distributions over the LDCT images throughout the data-driven learning.

Three subcategories of deep learning-based LDCT restoration techniques, namely discriminative, generative, and hybrid (generative and discriminative) can be distinguished according with the network model used. The discriminative network models represent bottom-up execution to separate learnt data based on a decision boundary. It has been discovered that the racially discriminatory models employed in LDCT restoration use Convolutional Neural Networks (CNN) and their variants, such as us Stacked Competitive Network (SCN) (Du et al., 2017), Residual Network (ResNet) (Kaiming He, Xiangyu Zhang, Shaoqing Ren, 2006), VGG19 (Simonyan & Zisserman, 2015). The MSE-based objective function, however, causes the structures to be too smoothed. Also, the outcomes of the ResNet-based investigations have been diminished because they cannot be generalized (Kulathilake et al., 2021).

The probabilistic distribution of the data is determined by deep learning models grouped as part of the generative process. The top-down execution of the generative approach is evident when compared to the discriminative approach. Additionally, it uses the unsupervised learning

approach to discover features. The most popular generating models for LDCT restoration were found to be the autoencoder and U-net models (Kulathilake et al., 2021).

In the hybrid learning approach, the learning model is built using both generative and discriminative network models. This hybrid learning technique has gained popularity in LDCT restoration following the introduction of GAN by Goodfellow et al. (B. I. Goodfellow et al., 2014). The generator for medical image denoising creates samples by learning the distribution of low-dose medical images. The discriminator receives both the synthetic images created by the generator and the regular dosage images, and it attempts to tell them apart (Creswell et al., 2018). GAN is flexible to implement different generator models based on various CNN architectures, such as the encoder–decoder (Ding, Long, Zhang, & Fessler, 2018), U-Net (Park, Baek, You, Choi, & Seo, 2019), and ResNet (Hu et al., 2019). Also, the discriminator mostly acts as a binary classifier to distinguish the synthetic and NDCT images apart. Depending on the adversarial learning method and the objective function used, several variants of GAN architectures have recently been used by researchers to denoise images, and the results have been astounding. As a result, denoising will undoubtedly improve with good GAN architecture.

## 2.2. Generative Adversarial Network

In 2014, GAN (B. I. Goodfellow et al., 2014) was proposed by Goodfellow. Generator G and Discriminator D are two stand-alone interacting neural networks that are trained cooperatively by playing a zero-sum game in which the generator attempts to distinguish between actual and created input.

The two interacting networks, the generator, and discriminator have fought each other in an adversarial way, as we tried to indicate in the previous section. As a result, the discriminator intends to separate fake samples from real input data. The generator network, on the other hand, is all about generating fake samples that are as good as the actual ones. Both models adjust their weight during training to improve until they reach Nash–equilibrium or zero-sum.

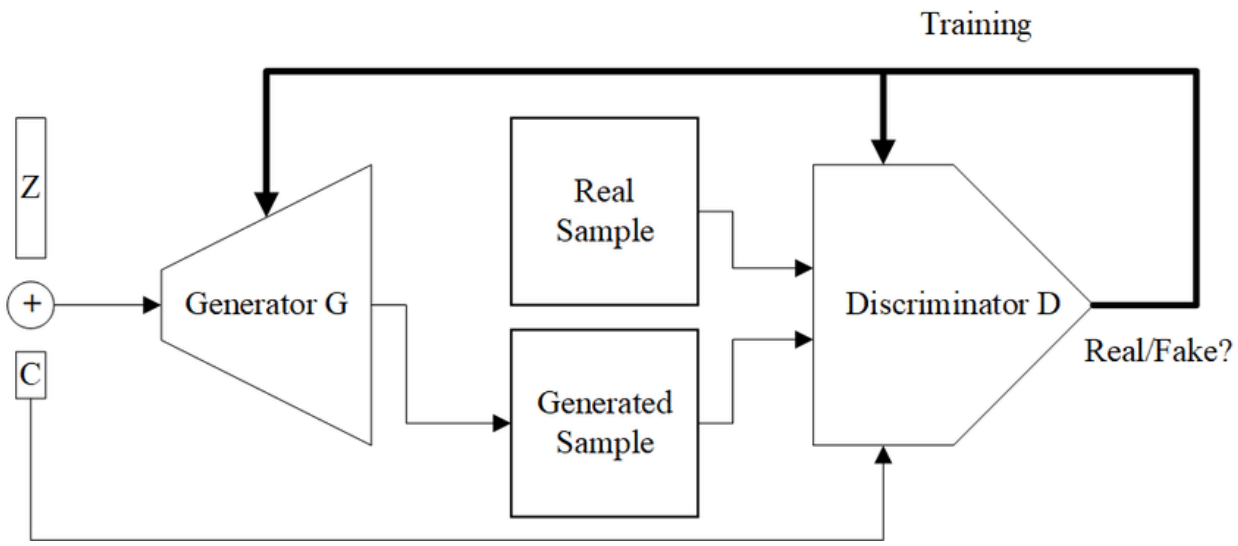



Figure 2.1 GAN framework structure

Source: adapted from (Öngün & Temizel, 2019)

### 2.2.1. GAN Training

Machine learning is all about generalization, which involves the model learning from real-world data in order to effectively predict the test set. GAN is no different, but unlike many other deep learning models, training is a bit difficult (Hitaj, Ateniese, & Perez-Cruz, 2017). So let us start with understanding how the adversarial conflict is performed between the discriminator and generator.

A discriminator network is a type of classifier that tries to separate real data from made-up data. Real data distribution is used as a positive example during training, while manufactured data distribution is used as a negative example.

- The loss function of GAN training:

GANs are algorithms that attempt to imitate a probability distribution. As a result, they should apply loss functions that reflect the distance between the GAN-generated data distribution and the distribution of the real data (I. Goodfellow et al., 2020). There are two loss functions that can be used in a GAN: one for generator training and the other for discriminator training. The generator and discriminator losses in the loss schemes are derived from a single measure of distance

between probability distributions. However, the generator can only change one component in the distance measure in any of these schemes: the term that reflects the distribution of the fake data.

Let us see the common GAN loss function here:

a) **minimax loss**: The loss function used in the paper that introduced GANs (B. I. Goodfellow et al., 2014).

$$E_x[\log(D(x))] + E_x[\log(1 - D(G(z)))] \tag{2.1}$$

Since their inception, GANs have been regarded as difficult to train: most of the time, the generator is incapable of producing anything helpful or repeatedly creates the same thing (mode collapse).

### 2.2.2. Least Squares GAN (LS-GAN)

The discriminator is hypothesized as a classifier with the sigmoid cross-entropy loss function in regular GANs. During the learning process, however, this loss function may cause the vanishing gradients problem. Xudong Mao (Mao et al., n.d.) devised the Least Squares Generative Adversarial Networks (LSGANs), which use the least-squares loss function as the discriminator to solve this problem.

Assume the discriminator uses an a-b coding scheme, with a and b representing the labels for fake and real data, respectively.

Then, for LSGANs, the objective functions are as follows:

$$\min_D V_{LSGAN}(D) = \frac{1}{2}E_{x\sim pdata(x)}[(D(x) - b)^2] + \frac{1}{2}E_{z\sim p_z(z)}[(D\big(G(z)\big) - a)^2] \tag{2.2}$$

$$\min_G V_{LSGAN}(G) = \frac{1}{2}E_{z\sim p_z(z)}[(D\big(G(z)\big) - c)^2 \tag{2.3}$$

where c denotes the value that G wants D to believe for fake data.

GAN applications cover a wide range of tasks, including an image to image translation, image denoising, and video production, to name a few.

### 2.2.3. Low Dose CT Imaging

CT is an X-ray treatment that involves computer processing to create 2D or 3D cross-sectional images, and it is one of the most important imaging modalities in today's hospitals and clinics. CT scans reveal anatomical shape, size, density, and internal abnormalities in greater detail than standard X-rays (Diwakar & Kumar, 2018). However, the patient may be exposed to radiation, as x-rays have the potential to cause genetic damage and cancer in a proportional manner to the radiation dose.

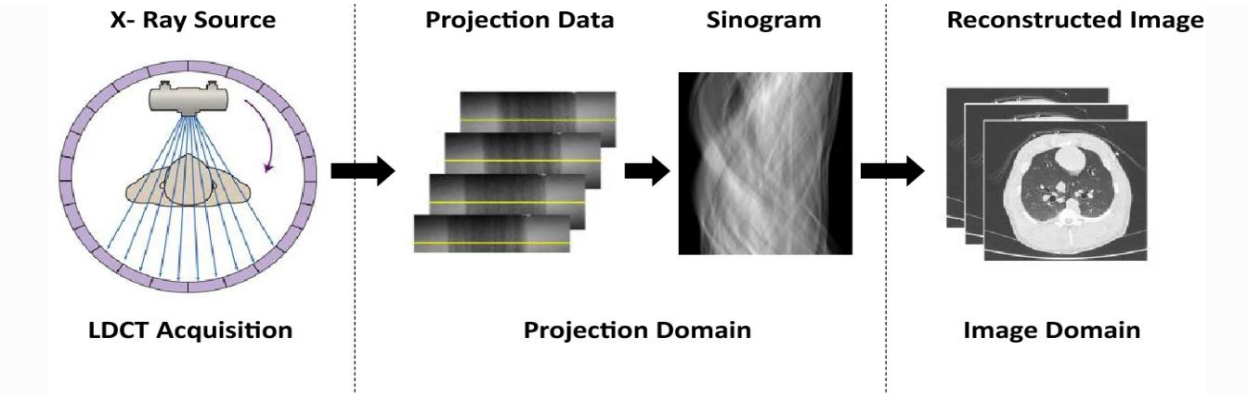



Figure 2.2 A diagram of the CT imaging

Source: adapted from (Kulathilake et al., 2021)

A low-dose CT scan is a quick, painless, and non-invasive screening method that uses the least amount of radiation possible. The low-dose CT scan emits about five times less radiation than a standard CT. A low-dose CT scan normally exposes the patient to 1-4 millisieverts of radiation, depending on their size whereas, normal conventional CT scan produces 5 to 20 millisieverts. Reconstructed images with lower radiation doses have more noise and artifacts, which can compromise diagnostic information. As a result, much work has gone into developing improved image reconstruction or image processing methods for low-dose CT.

## 2.3. LDCT Noise

During LDCT acquisition, CT images are generally damaged by quantum noise and other artifacts. Because of the X-ray photon deprivation during image capture, quantum noise is encoded in LDCT (Diwakar & Kumar, 2018). Disconnecting the edges, smoothing the target

subtle structures, and forming the low-contrast visuals due to the lack of X-ray photons are the visual degradations of quantum noise. During experiments, the quantum noise is usually approximated by a Poisson distribution (Diwakar & Kumar, 2018). In some cases, the noise distribution of CT images is calculated using the Mixed Poisson Gaussian distribution (MPGD) (Kaur et al., 2018). The Gaussian and Poisson distributions will be used to model both the electrical and quantum noise components in MPGD (Ding et al., 2018).

a, Normal dose abdomen CT image b**,** Quantum noise corrupted abdomen CT image

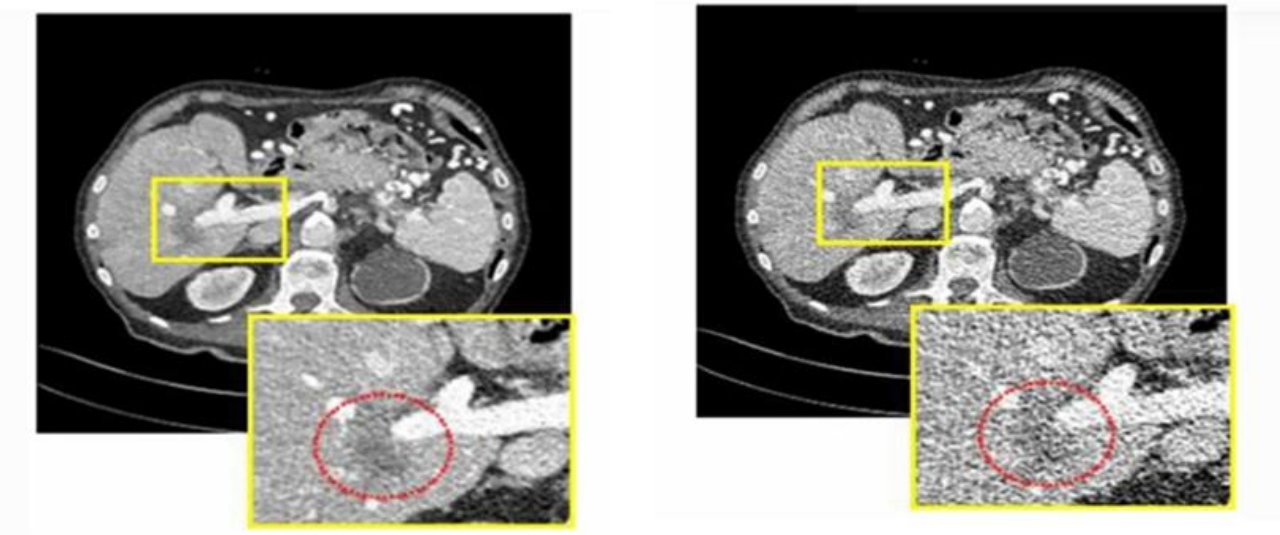


Figure 2.3 Visuals of CT Degradations.

Source: adapted from (Yang et al., 2018)

### 2.3.1. LDCT Noise Simulation

The dominant noise of a low-dose CT image in the projection domain has a Poisson distribution, according to the literature (Conolly, Nishimura, & Macovski, 1986), (J. Wang et al., 2008). As a result, Poisson noise is applied to the sinogram data of the normal-dose CT image to create a low-dose CT image. This approach is demonstrated in the steps below (Supanich et al. 2013):

a) Compute the Hounsfield unit numbers of the normal dose CT image H $\cup_{nd}$ from its pixel values, using the Eq. 2.4 (if the CT image has padding, it should be removed, first),

$$H \cup_{\mathrm{nd}} = \frac{PixelValue}{\mathrm{Slope}} + Intercept \qquad (2.4)$$

Slope and Intercept values can be found from DICOM header.

b) Compute the linear attenuation coefficients $\mu_{nd}$ based on water attenuation $\mu_{water}$,

$$\mu_{nd} = \frac{\mu_{water}}{1000} H \cup_{nd} + \mu_{water} \quad (2.5)$$

c) Obtain the projection data for normal-dose image $\rho_{nd}$ by applying radon transform on linear attenuation coefficients. To eliminate the size factor, this should be multiplied by the voxel size.

d) Compute the normal-dose transmission data $T_{nd}$,

$$T_{nd} = \exp(\rho_{nd}) \quad (2.6)$$

e) Generate the low-dose transmission $T_{ld}$ by injecting Poisson noise,

$$T_{ld} = \text{Poisson}\ (I_{ld}^{\circ} T_{nd}) \quad (2.7)$$

here, $I_{ld}^{\circ}$ is simulated low-dose scan incident flux.

f) Calculate the low-dose projection data $\rho_{ld}$,

$$\rho_{ld} = \ln(\frac{I_{ld}^{\circ}}{T_{ld}}) \quad (2.8)$$

g) Find the projection of the added Poisson noise,

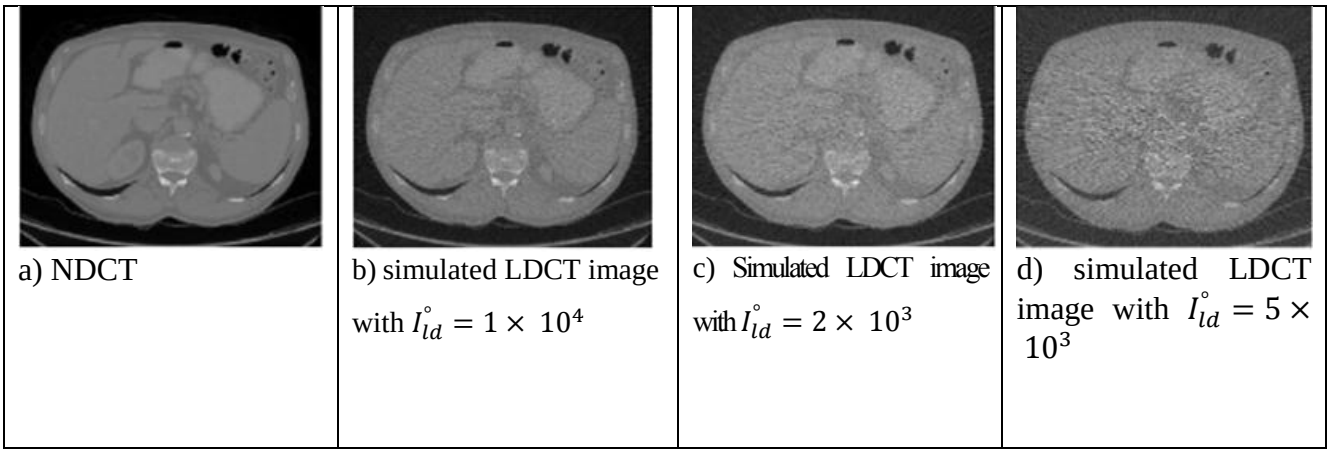
$$\rho_{noise} = \rho_{nd} - \rho_{ld} \quad (2.9)$$

h) Compute the linear attenuation of the low-dose CT image $\mu_{ld}$,

$$\mu_{ld} = \mu_{nd} + \text{iradon}\left(\frac{\rho_{noise}}{voxel}\right) \quad (2.10)$$

where, iradon represents the inverse Radon transform.

i) Finally apply the inverse of Eq. 2 to find the Hounsfield unit numbers for the low-dose CT image. Figure 2.4 demonstrates a normal-dose image and the simulated low-dose images with different incident flux $I^{\circ}$.



a) NDCT | b) simulated LDCT image with $I_{ld}^{\circ} = 1 \times 10^4$ | c) Simulated LDCT image with $I_{ld}^{\circ} = 2 \times 10^3$ | d) simulated LDCT image with $I_{ld}^{\circ} = 5 \times 10^3$

Figure 2.4 Simulation of a low-dose CT image from an upper abdominal CT image

Source: adapted from (Gholizadeh-Ansari, Alirezaie, & Babyn, 2019)

### 2.3.2. LDCT Denoising Algorithms

Over the last few decades, many LDCT restoration methods have been presented, all of which can be divided into three categories.

a) **Projection Filtering/ Sinogram Domain Filtering** (Paris, n.d.), (Balda et al., 2012), (Ouyang et al., 2011) These approaches operate directly on the raw projection data generated prior to back-projection. As a result, the restoration methods are fast and precise at computing noise statistics. Sinogram domain filtering approaches include structural adaptive filtering, bilateral filtering, and the penalized likelihood method. The advantage of these forecast statistics is that they are more computationally efficient however, are vendor-specific and not available to the general public. In addition, the sinogram domain filtering-restored LDCT images always result in a loss of spatial resolution and image edge blur.

b) **Iteration Reconstruction** (Q. Xu, Yu, Member, Mou, & Zhang, 2012), (Yi Zhang, Xi, et al., 2016), (Cai et al., 2014), (Liu, Ma, Fan, & Liang, n.d.). They primarily rely on the prior information in the image and accomplish noise reduction by iterating between the sinogram and image domains. Some of the priors utilized in the iterative reconstruction-based restoration category are non-local means (Yuan et al., 2018) total variation(Yi Zhang, Wang, et al., 2016), dictionary learning (Q. Xu et al., 2012), and low-rank approximation (Cai et al., 2014). Even though this LDCT restoration category produces attractive CT enhancement outcomes in terms of increasing the signal-to-noise ratio, iterative reconstruction-based CT restoration has been observed to have substantial computing costs and content loss.

c) **Image domain restoration** (Z. Li et al., 2014), (Feruglio, n.d.). They have been implemented into the CT imaging system and analysis system since they may be performed directly on the images and have lower calculating costs. Non-local means, total variation, Block Matching Three dimensions (BM3D), and statistics-based algorithms are examples of well-known image denoising techniques. Even if the image domain restoration methods are flexible enough to be implemented, the inability to compute the noise statistics due to its non-uniformity would restrict the accuracy of the proposed CT restoration applications. As a result, the present LDCT restoration methods and their limitations have paved the way for fresh LDCT restoration methods to be proposed.

Among the above-listed algorithms image domain restoration algorithms, which are indexed as Post-processing methods, are preferred since they can be directly applied to reconstructed images rather than raw data. Deep Learning (DL)-based LDCT restoration systems are considered Post-processing methods and have been increasingly popular in recent years, as opposed to traditional LDCT restoration methods, due to their data-driven, high-performance, and quick execution properties (Tian, Jia, Yuan, & Pan, 2011).

The CT reconstruction method generally entails mapping features of normal-dose CT (NDCT) images to low-dose CT (LDCT) images, which is accomplished using denoising techniques. The learning process includes two major components: network architecture and loss function. The architecture determines the complexity of the denoising model and the loss function controls what the denoising model learns (Y. Chen et al., 2009).

## 2.4. Related Work

LDCT denoising has been studied by several scholars throughout the years. We looked at the works that were thought to have a strong connection to our thesis.

Chen et al. (H. Chen, Zhang, Zhang, et al., 2017) were the first to propose a low-dose CT image denoising method based on convolution neural networks (CNN), which produced better visual sense and measurement results. The network structure was then refined by Chen et al. (H. Chen, Zhang, Kalra, et al., 2017), who created a residual encoder CNN (RED-CNN). The results were superior to those of CNN. However, the LDCT images were over-smoothed and blurred as a result of the MSE loss, which tends to average the findings, resulting in the loss of structural information. Their network, on the other hand, was complicated and time-consuming.

(Gu & Ye, 2021) proposed a cycleGAN with a U-net generator with 10 convolution layers and Adaptive Instance normalization (AdaIn) layers, and a Patch-GAN structure with 6 convolution layers as a discriminator for LDCT denoising. However, this approach has an issue with Structure preservation so it needed to be improved further.

WGAN-VGG (Yang et al., 2018) employed a transfer learning approach by implementing pre-trained VGG-19 perceptual loss in the WGAN model. However, the VGG feature extractor is pretrained on a natural image dataset that does not contain medical images, using it as a perceptual loss results in some distorted images.

Another fine work by Babyn et al. (Yi & Babyn, 2018) proposes an LDCT denoising model using UNet256 as a generator, and Patch-GAN as a discriminator. Even tough, it introduces Sharpness-Aware method Using Conditional GAN for Low-Dose CT Denoising to reduce a blur effect on the final reconstructed results, In some low-contrast areas, the sharpness metric's sensitivity is minimal.

LDCT denoising with novel 3D self-attention convolutional neural network (M. Li et al., 2020) further extend the state of the art in the area of LDCT by adding attention module leverages the 3D volume of CT images to capture a wide range of spatial information both within CT slices and between CT slices. However, Low PSNR and SSIM for some cases due to not using the pixel wise loss function.

In (Image et al., 2021) the authors propose DRWGAN-PF a generative adversarial network that incorporates both multi-perceptual and fidelity loss. By reducing the difference between the LDCT image and the normal-dose computed tomography (NDCT) image in the feature space, multi-perceptual loss employs the image's high-level semantic characteristics to achieve the goal of noise suppression. However, perceptual loss is determined using natural images (VGG-19 net) this results in some distorted images.

A new network was proposed by Yang et al. (You et al., 2018), with 8 convolution layers as a generator, CNN with 6 convolutional layers as a discriminator, specifically incorporate 3D volumetric information to improve the image quality. However in several situations, edge blurring and structural dissimilarities appeared.

A novel method for denoising LDCT images was proposed by (Huang et al., 2022). The authors suggested dual domain U-net based discriminator they so called as DU-GAN. For denoised LDCT image generation, the model learns by image to image translation from NDCT to LDCT using Residual Encoder-Decoder CNN with 10 convolution and deconvolution layers. Dual U-net based discriminators are used to learn both global and local difference between the denoised and normal dose images in both image and gradient domains. Although it came up with a better approach, blurring and degradation in structural features occurs due to the use of MSE loss as pixel-wise loss and a lack of attention mechanism provided to the preservation of structural features respectively.

Task-oriented Low Dose CT denoising by Zhang (J. Zhang et al., 2021), was the first approach to coming up with a novel approach with a task-oriented denoising model and task-oriented loss leveraging knowledge from downstream tasks. The key disadvantage of this study is that the dataset used to train their model was projection data that was then combined with noise and reconstructed using CT simulation software (Divel & Pelc, 2019). These sinogram-based methods, on the other hand, are frequently time-consuming, and the simulated projection data may not accurately match real-world situations, therefore the simulated LDCT noise is likely to be imperfect and affects the training model.

## 2.5. Summary of Related Works

In earlier work, researchers designed various architectures to learn a mapping from an NDCT image to LDCT for learning the denoising process in an unsupervised manner, as indicated in the table below. However, because those efforts rely on different-GAN architectures to handle the denoising, the majority of them pay less attention to structural feature preservation and adopt better pixel-wise loss functions, that can handle the denoising process, during model training. In fact GAN-based LDCT denoising approaches often retain GAN competition at the structural level, with a discriminator that gradually down-samples the input into a scalar value. However, because the distribution of synthetic samples shifts when the generator changes throughout training, the discriminator is prone to forgetting earlier samples, failing to retain a powerful data representation to define the global and local picture difference, besides they did not consider the ability of the pixel-wise loss functions to tolerate those features added for improving the structure preservation while denoising (Kulathilake et al., 2021). Among the various issues

mentioned in the above section, this study attempts to address the problem that most prior works faced commonly, notably a lack of an attention mechanism to structural detail preservation and a pixel-wise loss function that can tolerate the significant amount of noise in the denoising process during model training, we can see this from the table below that some of the works try to use attention mechanism without considering the effect of what kind of pixel-wise loss they have used: on the other hand some works try to apply pixel-wise loss using perceptual loss such as VGG-19 net, which is pre-trained by natural images that are fundamentally different from medical images, and MSE loss, which mostly suffers from over smoothing and blurring issue. This paper proposes a technique for embedding an attention mechanism with adequate cross channel communication, which will enhance the structural detail preservation, and better pixel wise loss function using GAN to solve the above-stated problems. Table 2.1 illustrates a summary of related work.

Table 2.1 Summary of related works

| Related papers | Dataset used | Methodology | Limitation |
|---|---|---|---|
| (H. Chen, Zhang, Kalra, et al., 2017) | National Biomedical Imaging Archive (NBIA) | Residual Encoder Decoder-CNN with 5 convolution and deconvolution layers | ○ Due to the use of an objective function based on MSE, texture loss and blurring occurred.<br>○ There is a lack of attention mechanism provided to the preservation of structural features. |
| (Gu & Ye, 2021) | Cardiac multiphase Coro-nary CT (CTA) | U-net generator with 10 convolution layer and AdaIn layers, Patch-GAN structure with 6 convolution layers | ○ Lack of attention mechanism given to structural feature preservation |

| | | | |
|---|---|---|---|
| (Yang et al., 2018) | National Cancer Imaging Archive (NBIA). | WGAN with CNN with 6 convolution layers in the discriminator and VGG as a perceptual loss | ○ Natural images are used to determine perceptual loss (VGG-19 net).<br>○ Because the VGG feature extractor is pre-trained on a natural image dataset that does not contain medical images, using it as a perceptual loss results in some distorted images. |
| (Yi & Babyn, 2018) | National Cancer Imaging Archive (NBIA). | SAGAN-UNet256 as a generator, Patch-GAN as a discriminator | ○ In some low-contrast areas, the sharpness metric's sensitivity is minimal.<br>○ Lack of attention mechanism given to structural feature preservation |
| (M. Li et al., 2020) | National Cancer Imaging Archive (NBIA). | SA-CNN-CNN with eight convolution layer and three 3D self-attention blocks as a generator, CNN with 6 convolution layers | ○ Due to the lack of a pixel-wise loss function in some circumstances, PSNR and SSIM are low. |
| (You et al., 2018) | 2016 NIHAAPM-Mayo Clinic Low-Dose CT Grand Challenge | SMGAN 3D CNN with 8 convolution layers as a generator, CNN with 6 | ○ Edge blurring and structural dissimilarities appeared<br>○ There is a lack of attention mechanism provided to the preservation of structural features. |

| | | | |
|---|---|---|---|
| | | convolutional layers as a discriminator | |
| (Huang et al., 2022) | 2016 NIHAAPM-Mayo Clinic Low-Dose CT Grand Challenge | DU-GAN-Residual Encoder-Decoder CNN with 10 convolution and de-convolution layers as a generator, dual UNet based discriminators | ○ Blurring occurs due to the use of MSE loss as pixel-wise loss<br>○ There is a lack of attention mechanism provided to the preservation of structural features. |

# CHAPTER THREE

# 3. RESEARCH METHODOLOGY

## 3.1. Chapter Overview

This chapter grants the methodologies, procedures, and techniques used in achieving the main objective of the thesis work, improving the quality of denoised LDCT images. Datasets used, data augmentation approaches, preprocessing techniques, development tools, baseline work, and finally evaluation methods are all covered in this chapter.

## 3.2. Research Approach and Procedure of the Study

Given the nature of the problem at hand and the research objectives of the thesis, this study falls within the category of quantitative research, which may provide results through experimentation. In light of this, the techniques are research design techniques that include data gathering and analysis, data quantification, relationship construction, and mathematical models to describe phenomena or support a theory (Sam, 2012).

Several various methods have been used to achieve the research work's objective. To achieve the desired outcome of the planned work, those are a series of steps or actions. The main steps or procedures used in the operation of this thesis are identifying the research area, reviewing what has already been done and known by various researchers, gap investigation, research question formulation, tool and methodology selection, dataset preparation, and finally designing and implementing the best possible solution. The following figure depicts a great view of the actions.

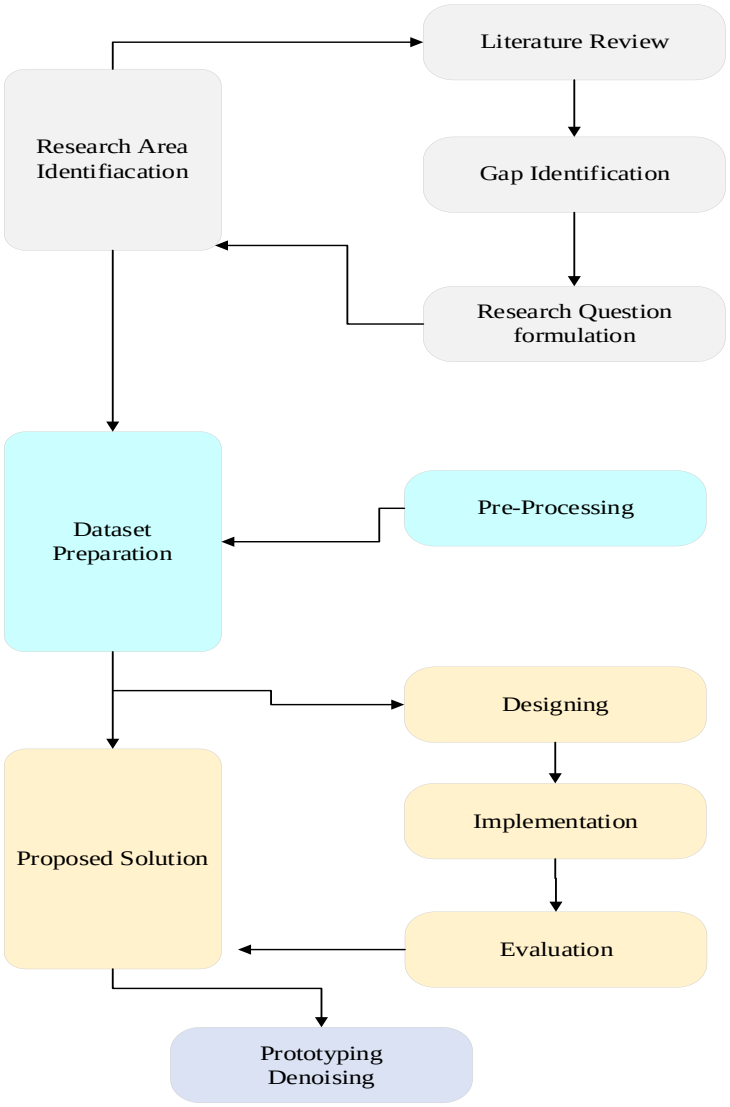



Figure 3.1 Procedures of the Research

## 3.3. Dataset

Deep learning algorithms rely significantly on data to learn and cannot learn without it. The suggested model was trained and evaluated on one dataset: The LDCT dataset, which was originally produced for the 2016 NIHAAPM-Mayo Clinic Low-Dose CT Grand Challenge.

### 3.3.1. NIHAAPM-Mayo Clinic Low-Dose CT

The LDCT dataset used in this work was a publicly available a real clinical dataset, containing CT projection data from patient exams, both at clinical doses and at simulated lower levels, which was originally created for the 2016 NIHAAPM-Mayo Clinic Low-Dose CT Grand

Challenge and was just released in (Moen et al., 2021). It offers scans from three areas of the body, each with a distinct simulated low dose: 25% for the head, 25% for the abdomen, and 10% for the chest. 10% chest datasets, named the Mayo-10% chest datasets were used in this research because 10% of the normal dose in the chest is more difficult to achieve than 25% of the normal dose in the abdomen and head, following (Huang et al., 2022) the NDCT scans of 20 anonymous patients and the equivalent simulated LDCT (1/4 dose) images after adding realistic noise are randomly selected for training and another 10 patients for testing for each dataset; there is no identity overlap between training and testing. In particular, 10,000 training and 6439 test images were chosen at random from each collection.

Among the dataset comprises of 10,000 images, for our experiments, a training set of 100,000 pairs of randomly selected image patches from 10,000 slices were used. The patch size was 64 × 64 as mentioned in section 3.5.2. Also, 6439 pairs of image patches were extracted for testing with full resolution 512x512.

LDCT NDCT

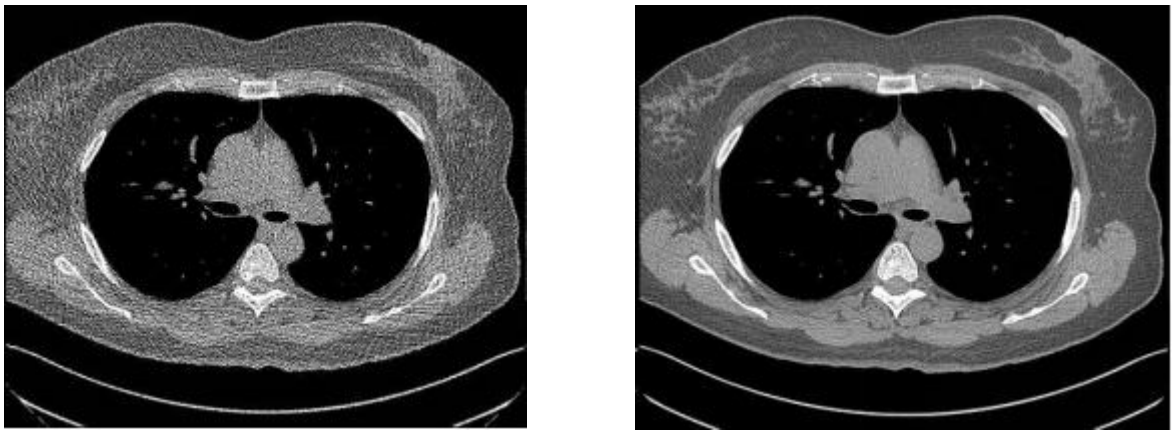


Figure 3.2 NIHAAPM-Mayo Clinic Low-Dose CT dataset sample examples

## 3.4. Data Augmentation

A training set can be enlarged via image data augmenting by adding significantly changed copies of current data or newly developed synthetic data from existing data. It functions as a regularizer and aids in the reduction of over-fitting when training a machine learning model. When fit models are trained on additional data, their ability to generalize what they've learnt to new images improves.

As training progresses, the discriminator's ability to recognize local variations between actual and fake samples decreases, which may adversely affect denoising performance. Furthermore, the discriminator is meant to concentrate on global structure changes and local features at the per-pixel level. To overcome these concerns, the CutMix augmentation strategy is used to regularize the discriminator, and can help the discriminator understand the intrinsic difference between real and false samples. This work adopts exactly the same CutMix augmentation setting in (Huang et al., 2022).

CutMix is a data augmentation strategy that tackles the issue of regional dropout algorithms' information loss and inefficiency (Yun, 2019). Rather than removing pixels and filling them with black or grey pixels or Gaussian noise, the deleted portions are replaced with a patch from another image, and the ground truth labels are mixed proportionately to the combined images' pixel count (Huang et al., 2022). It's implemented via the following formulas:

$$\tilde{x} = Mx_i + (1 - M)x_i \tag{3.1}$$

$$\tilde{y} = \lambda y_i + (1 - \lambda)y_i \tag{3.2}$$

Where M – is the binary mask which indicates the cutout and the fill-in regions from the two randomly drawn images and λ (in [0, 1]) is drawn from a Beta (α, α) distribution (it is a family of continuous probability distributions defined on the interval [0, 1])

## 3.5. Pre-processing Techniques

### 3.5.1. Normalization

The goal of normalization is to convert features to a common scale. This enhances the training stability and performance of the model. In this research, All images are linearly normalized to [0, 1] during training.

$$x = \frac{y - \mu}{\delta} \tag{3.3}$$

Where $y$ is the input image that has a value between 0 and 1, $\mu$, and $\delta$ are mean and variance respectively. Both mean and variance have 0.5 values.

### 3.5.2. Resize

Image resizing is a critical preprocessing step in deep learning that allows adjusting input data to the size of the network. Since techniques based on deep learning require a large number of samples. In practice, it is difficult to meet this condition, particularly for clinical imaging. To overcome this issue, in this work, overlapped patches in CT images that are 64x64 in size used. Due to the ability to detect regional perceptual variations and the greatly increased sample size, this technique has been demonstrated to be both effective and efficient (H. Chen, Zhang, Kalra, et al., 2017)(Huang et al., 2022). Therefore, the training images in the dataset were resized to $64x64$ which are then directly applied to the whole image i.e. with image size $512x512$ for visualization and testing.

## 3.6. Development Tools

### 3.6.1. Hardware Tools

In the table below, you'll find a list of the hardware tools that were used throughout the study.

Table 3.1 Hardware tools

| Number | Device Name | Usage for the research |
|---|---|---|
| 1 | Gpu | To train the model |
| 2 | Hard disk | To store the datasets and models |
| 3 | RAM | To improve hardware performance |

### 3.6.2. Software Tools

Different software and coding tools are utilized to implement the research through coding, and those are given below with description

**Anaconda**[5]: is a combination of Python and R that streamlines deployment and package management. N

Being able to provide a multitude of tools for machine learning and data analysis with just one installation makes Anaconda very well known. Conda, the package manager for Anaconda provides for the installation of libraries. It also features a useful interaction with pip, which

[5]https://www.anaconda.com/

permits the installation of libraries that are not accessible through Anaconda's package manager. It can also be used on Linux, macOS, and Windows because it is platform-independent.

**Jupyter Notebook**[6]: An open-source, web application that integrates software code, explanatory text, computational output, and multimedia resources into a single document. It is integrated with the anaconda environment and used for editing code and seeing results.

**PyTorch**[7]: An open source machine learning framework that accelerates the path from research prototyping to production deployment. It is based on the torch library, which is used in applications such as computer vision and natural language processing. PyTorch is nearly as easy to learn as the Python programming language because its syntax is similar.

**OpenCV**[8]: is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage. The library is cross-platform and free for use under the open-source Apache 2 License.

## 3.7. Baseline Work

The implemented model is compared to models that focus on low-dose CT denoising using GAN to validate its effectiveness. DU-GAN (Huang et al., 2022)is chosen as a baseline model based on the model performance and generated outcomes when compared to others.

The DU-GAN model consists of one generator and two discriminators. The generator is constructed as Residual Encoder decoder CNN (REDCNN). And U-Net based discriminators in both image and gradient domains. U-Net is a semantic segmentation system that consists of an encoder, a decoder, and numerous skip connections that copy feature-maps from the encoder to the decoder to maintain high-resolution features. It has proven state-of-the-art performance in a variety of semantic segmentation and image translation tasks. The authors of the baseline work emphasize that in the context of LDCT denoising, U-Net and its variants are solely utilized as a denoising model and have not been investigated as a discriminator. DUGAN model use the U-Net to replace the typical classification discriminator in GANs with a U-Net style discriminator that maintains both global and local data representation. U-net based discriminator in image domain can provide more informative feedback to the generator including both local per-pixel and global structural information.

[6]https://jupyter.org/
[7]https://pytorch.org/
[8]https://opencv.org/

## 3.8. Evaluation Matrices

Evaluation metrics are used to measure the quality of the model and to make sure that the model is performing optimally and correctly. The model's quality will be assessed using measures such as the peak signal-to-noise ratio (PSNR), the structural similarity index measure (SSIM), and the root mean square error.

### 3.8.1. Peak Signal-to-Noise Ratio (PSNR)

It is a technical term for the ratio of a signal's maximal power to the power of corrupting noise that degrades the accuracy of its representation. Due to the high dynamic range of many signals, PSNR is typically stated as a logarithmic number using the decibel scale.

PSNR is most easily defined via the mean squared error (MSE). Given a noise-free $m \times n$ monochrome image *I* and its noisy approximation *K*, MSE is defined as

$$\text{MSE (x,y)} = \sum_{i-0}^{m-0} \sum_{j-0}^{n-0} \left[I(i,j) - K[i,j]\right]^2 \tag{3.4}$$

The PSNR (in dB) is defined as:

$$\text{PSNR} = 10.\log_{10} \frac{(MAX_I^2)}{MSE} \tag{3.5}$$

$$= 20.\log_{10} \frac{MAX_I}{\sqrt{MSE}}$$

$$=20.\log_{10}(MAX_I) - 10.\log_{10}(MSE)$$

Here, $MAX_I$ is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255. More generally, when samples are represented using linear PCM with *B* bits per sample, $MAX_I$ is $2^B - 1$.

### 3.8.2. Structural Similarity Index Measure (SSIM)

The SSIM is a complete reference metric, which means that image quality is evaluated or forecasted using a reference image that is original, uncompressed, or free of distortion. It is a perceptual paradigm that includes essential perceptual phenomena like brightness and contrast masking as well as image degradation as a perceived modification in the informational structure.

The assumption that pixels contain significant interdependencies, particularly when they are geographically close, is referred to as structural information. These dependencies include important details regarding the object structure of the visual scene.

Contrast is a phenomenon in which the visual distortions become less obvious in areas of intense activity in the scene as opposed to luminance, which is a phenomenon in which the visual This evaluation metrics used to measure similarities between two images.

Luminance, $$L(x,y) = \frac{(2\mu x\mu y + c1)}{(\mu x^2 + \mu y^2 + c1)} \quad (3.9)$$

Where $c1 = (k1 + l)^2$ is non-negative regularization constant used to avoid instability when the mean is close to zero, L is the dynamic range value, $k1$ is small constant value.

Contrast, $$C(x,y) = \frac{(2\sigma xy + c2)}{(\sigma x^2 + \sigma y^2 + c2)} \quad (3.10)$$

Where $c2 = (k2 + l)^2$ is nonnegative regularization constant used to avoid instability when the standard deviation is close to zero, $k2$ small constant value.

Structure, $$S(x,y) = \frac{(\sigma xy + c3)}{(\sigma x^2 + \sigma y^2 + c3)} \quad (3.11)$$

Where, μx average mean of x, μx average mean of y, $\sigma_{x^2}$ variance of x, $\sigma_{y^2}$ variance of y and $\sigma_{xy}$ covariance of x and y, and c3 = c2/2

$$SSIM(x,y) = [l(x,y)]^{\alpha}[C(x,y)]^{\beta}[S(x,y)]^{\gamma} \quad (3.12)$$

Where $\alpha$, $\beta$, and $\gamma$ are parameters that are used to adjust the relative importance of the three components.

### 3.8.3. Root Mean Squared Error

The root-mean-square error (RMSE) is a commonly used metric for comparing predicted and observed values (sample or population values) by a model or estimator. The square root of the average of squared mistakes is the RMSE. Each error's effect on RMSE is proportional to the squared error's size; consequently, larger errors have a disproportionately large effect on RMSD.

RMSE metric is given by:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Predicted_i - Actual_i)^2}{N}} \quad (3.13)$$

# CHAPTER FOUR

# 4. PROPOSED ARCHITECTURE

## 4.1. Chapter overview

This chapter explains the suggested architecture for improving the quality of denoised low dose CT scan images, including the necessary diagrammatic representation and mathematical expression. The first part of this chapter explains the general architecture of the proposed model to denoise LDCT into NDCT; the second section of this chapter describes the overall architecture of the proposed model to denoise LDCT into NDCT. The generator and discriminator architecture are discussed in the second section. The model learning functions are discussed in the third section, and finally, the training pseudo code is explained.

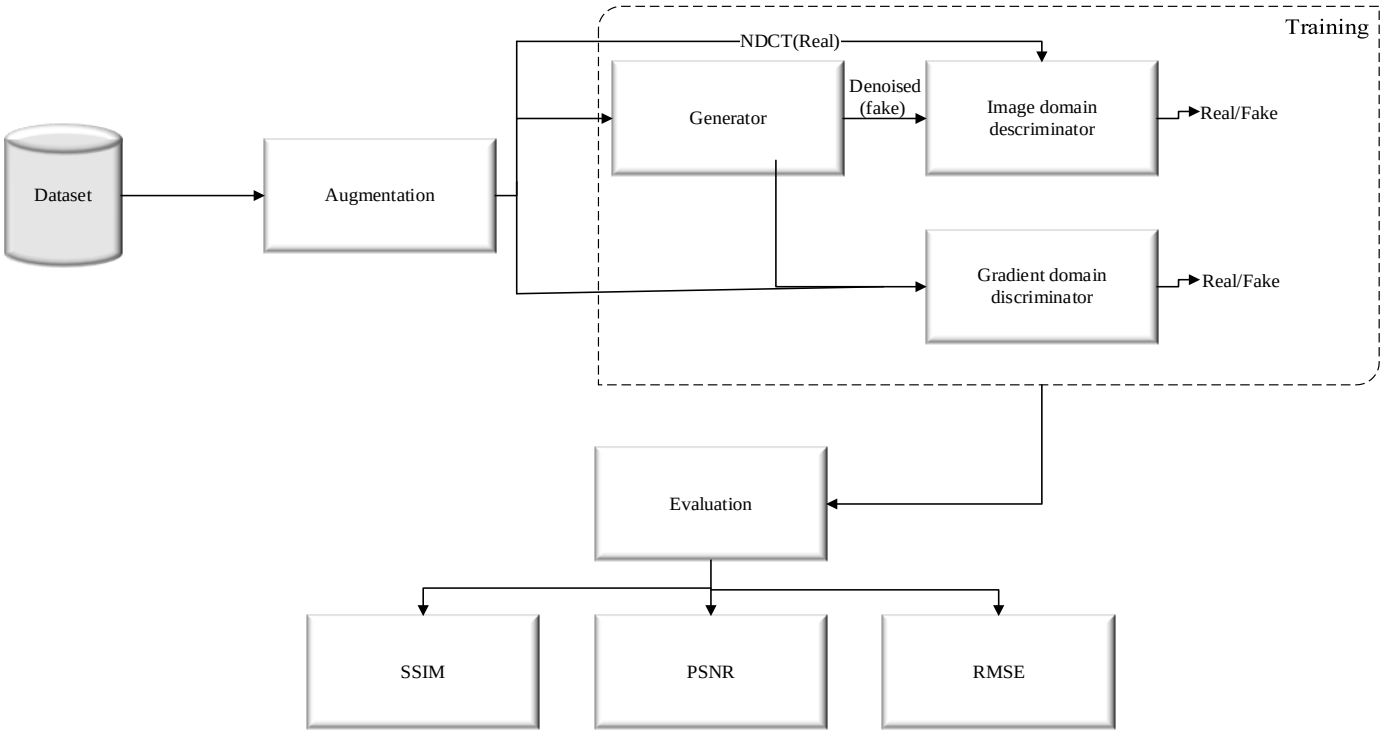



Figure 4.1 The proposed model process flow

## 4.2. Model Architecture

The proposed approach, entitled "ECADU-GAN," is used to generate denoised low dose CT images. For learning image-to-image translation, the suggested model was adapted from "DU-GAN: Generative Adversarial Networks with Dual-Domain U-Net Based Discriminators for Low-Dose CT Denoising." The generator network now has instance normalization, Efficient channel attention block, and dice loss as a pixel-wise loss in the image domain discriminator.

One generator and two discriminators are used in the proposed model. The RED-CNN (H. Chen, Zhang, Kalra, et al., 2017) architecture was used to construct the generator network. The discriminator networks are a U-Net discriminators that learns both global and local differences between denoised and normal-dose images in both image and gradient domains, as described in (Shinde, 2018).

Adjustments have been made to the generator network, and additional losses have been applied to the image domain discriminator; Section 5.6 addressed the depth implementation detail of the model architecture.

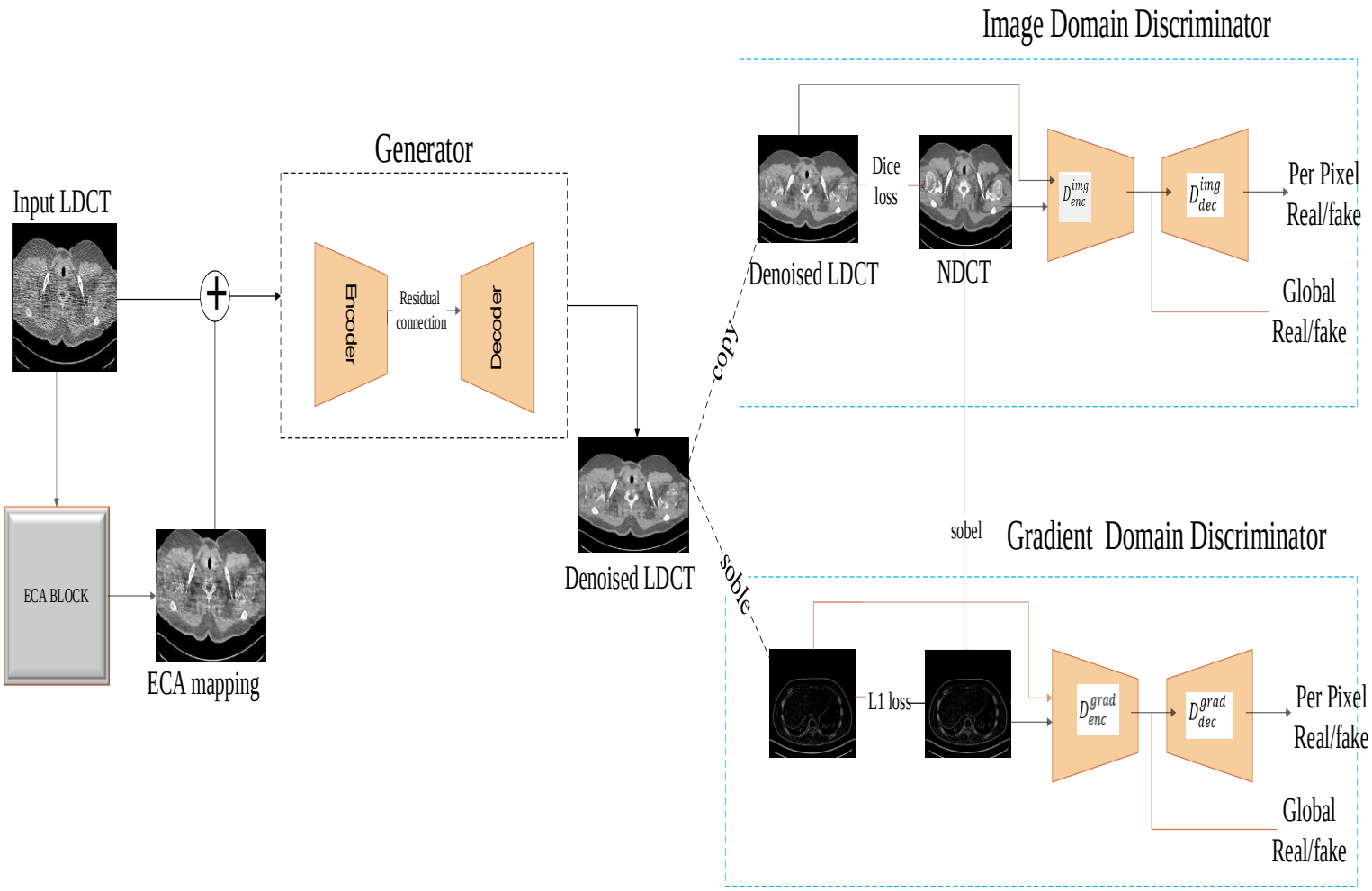



Figure 4.2 Overall Framework of the Proposed Solution

The encoder, decoder, and residual connection between the layers in the encoder and layers in the decoder make up the generator, as shown in figure 4.2. The encoder, which is a series of fully-connected convolutional layers employed as the stacking encoder, accept the LDCT image concatenated with the output attention map created by the attentive block, or efficient attention block, as input. The encoder is used to gradually reduce image noise and artifacts from low to high levels in order to protect the extracted patch's critical information.

The efficient channel attention block is used to provide cross channel communication and it aims at appropriately capturing local cross-channel interaction, so the coverage of interaction (i.e., kernel size k of 1D convolution) needs to be determined. To do this during the inference stage, we need estimate a mapping φ from the input LDCT without corresponding NDCT. Assuming that the coverage of interaction (i.e., kernel size k of 1D convolution) is proportional to channel dimension C. In other words, there may exist a mapping φ between k and C:

$$C = \varphi(k) \tag{4.1}$$

The simplest mapping, denoted by the formula in equation (4.9), is a linear function.

$$\varphi(k) = \gamma * k - b \tag{4.2}$$

But linear function-based relationships are too constrained. On the other hand, it is generally known that the power of 2 is typically specified for channel dimension C (i.e., the number of filters). Therefore, by converting the linear function in equation (4.2) to a non-linear one, i.e.,

$$C = \varphi(k) = 2^{(\gamma * k - b)} \tag{4.3}$$

Afterward, kernel size k can be adaptively chosen by using given channel dimension C.

$$k = \Psi(C) = \left|\frac{\log_2 C}{\gamma} + \frac{b}{\gamma}\right| \tag{4.4}$$

By adopting a non-linear mapping $\Psi$, (Q. Wang et al., 2020) it is evident that high-dimensional channels interact over longer distances whereas low-dimensional channels interact over shorter distances.

Finally, as mentioned before the mapping from the attentive block concatenated with input LDCT and fed to the generator encoder. As the result, the injected attention map φ would serve

as the prior information to guide the network in order to direct the removal of noise and preservation of structural detail, in the generator and discriminator.

The discriminator networks are utilized to contrast the denoised LDCT from that of the real NDCT using UNet based model both in image and gradient domain.

## 4.3. Generator Architecture

A learning generator that captures the training data distribution and generates new samples that are indistinguishable from the real ones. The generator's goal in LDCT denoising is to produce photo-realistic denoised results in order to mislead the discriminator(Yi & Babyn, 2018). The U-Net architecture is used in RED-CNN, however the down-sampling and up sampling processes are removed to prevent data loss. At both the encoder and decoder, the model has 10 convolutional and de-convolutional layers, each having 32 filters to reduce computation costs, and a ReLU activation function. There are ten remaining residual connections in all.

### 4.3.1. Instance Normalization

The normalization method, on the surface, appears to eliminate instance-specific contrast information from the content image, making generation easier (Z. Xu, Yang, Li, & Sun, 2018). This demonstrates how a minor tweak in the stylization architecture can result in a considerable improvement in the quality of the generated images.

Let $x \in R^{N\times C\times H\times W}$ denote the feature map of a convolutional layer derived from a mini-batch of image samples, where N is the batch size, C is the layer width (number of channels), and The feature map's height and width are H and W. The element at height h and width w and the $c^{th}$ channel from the nth sample is denoted by $x_{nchw}$. The instance normalization layer can be stated as,

$$IN(x_{nchw}) = \gamma_{nc}\left(\frac{x_{nchw} - \mu_{nc}}{\sigma_{nc}}\right) + \beta_{nc} \tag{4.5}$$

### 4.3.2. Efficient Channel Attention

The channel attention mechanism has recently been shown to have a lot of promise in terms of boosting the performance of deep convolutional neural networks (CNNs) (Q. Wang et al., 2020). Most existing techniques, on the other hand, focus on creating more sophisticated attention modules in order to improve performance, which necessarily increases model complexity. To

overcome the paradox of performance and complexity trade-off, (Q. Wang et al., 2020) proposes an Efficient Channel Attention (ECA) module with only a few parameters and a noticeable performance improvement. They show that avoiding dimensionality reduction is critical for learning channel attention, and that suitable cross-channel interaction can preserve performance while drastically reducing model complexity by dissecting the channel attention module in SENet (Squeeze and Excitation Network). As a result, a local cross-channel interaction technique that does not require dimensionality reduction and can be performed efficiently using 1D convolution.

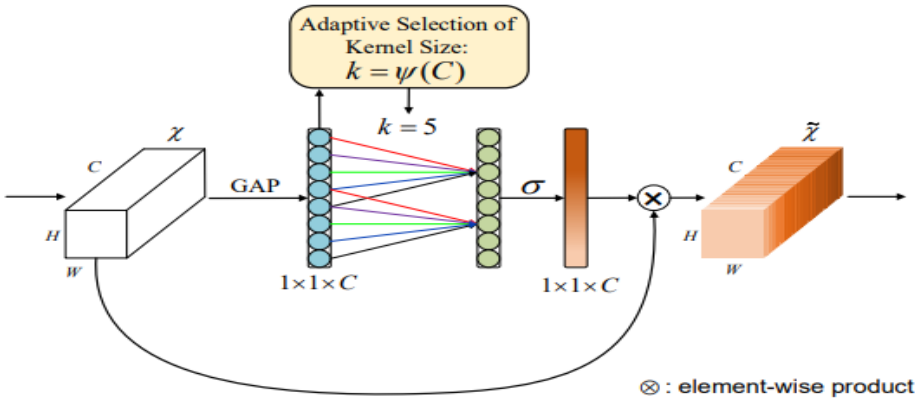



Figure 4.3 A Diagram for Efficient Channel Attention (ECA) Module.

Source: adapted from (Q. Wang et al., n.d.)

Given the aggregated features obtained by global average pooling (GAP), ECA generates channel weights by performing a fast 1D convolution of size k, where k is adaptively determined via a mapping of channel dimension C.

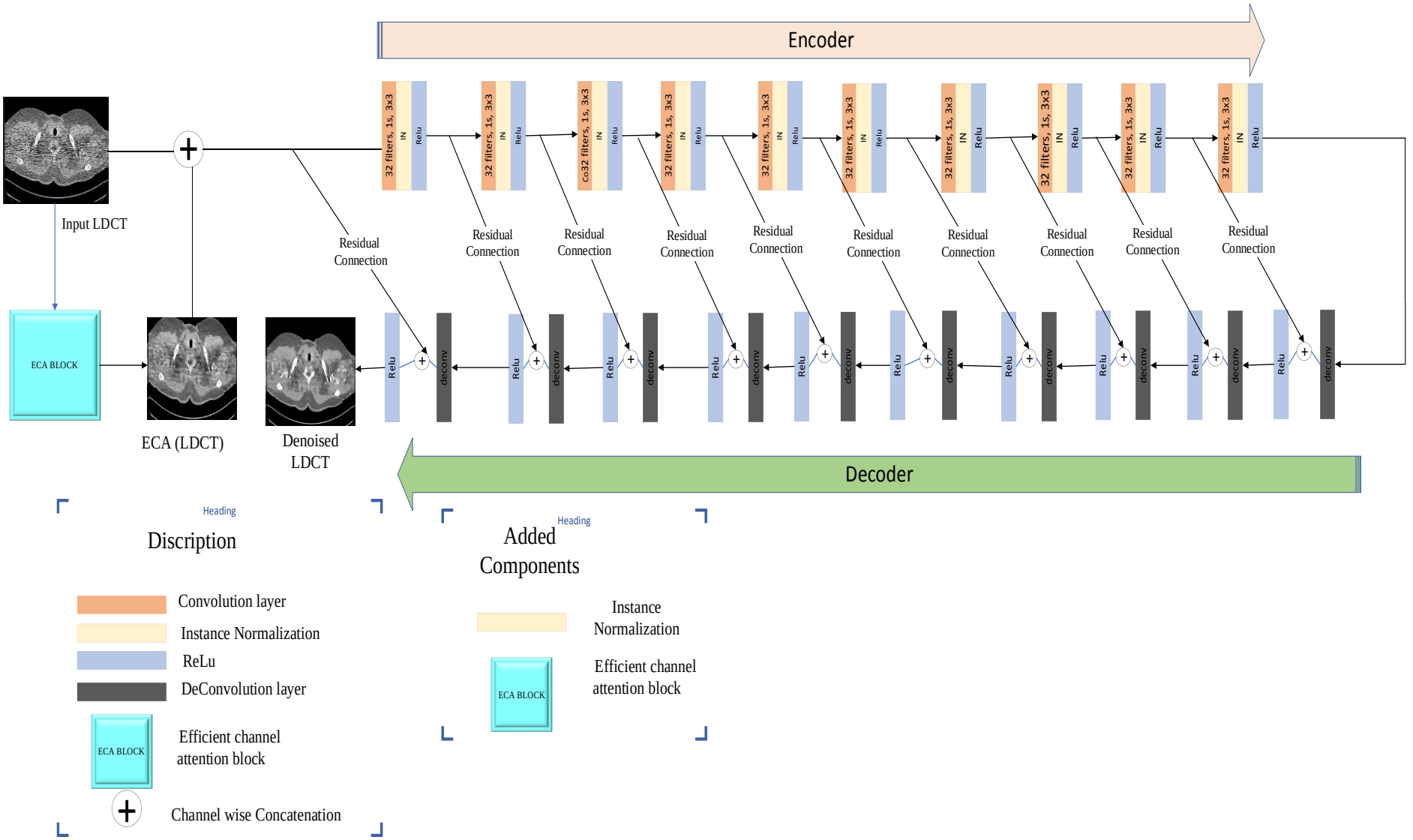



Figure4.4 Generator Architecture

## 4.4. Discriminator Network

The discriminator networks are a UNet based discriminators which can simultaneously learn the global and local difference between the denoised and normal-dose images in image and gradient domains. Combining the U-Net based discriminators in the image and gradient domains, two independent GANs competitions are maintained during training (Huang et al., 2022). The generator is responsible for denoising an LDCT image, which is subsequently fed into two independent discriminators in the image and gradient domains. In the image domain branch, the discriminator $D_{img}$ penalizes the generator for producing photo-realistic denoised LDCT, whereas in the gradient domain branch, the discriminator $D_{grad}$encourages better edge while reducing streak artifacts due by photon shortage. Additionally, each branch's discriminator uses a U-Net-based design to encourage the generator to focus on both global structure and local details, which can improve the denoising process' interpretability with the per-pixel confidence map output.

Most importantly, a spectral normalization layer (Miyato, Kataoka, Koyama, Yoshida, & Networks, 2018) and a Leaky ReLU activation with a slope of 0.2 for negative input follow each convolutional layer of Discriminator except the last one are added.

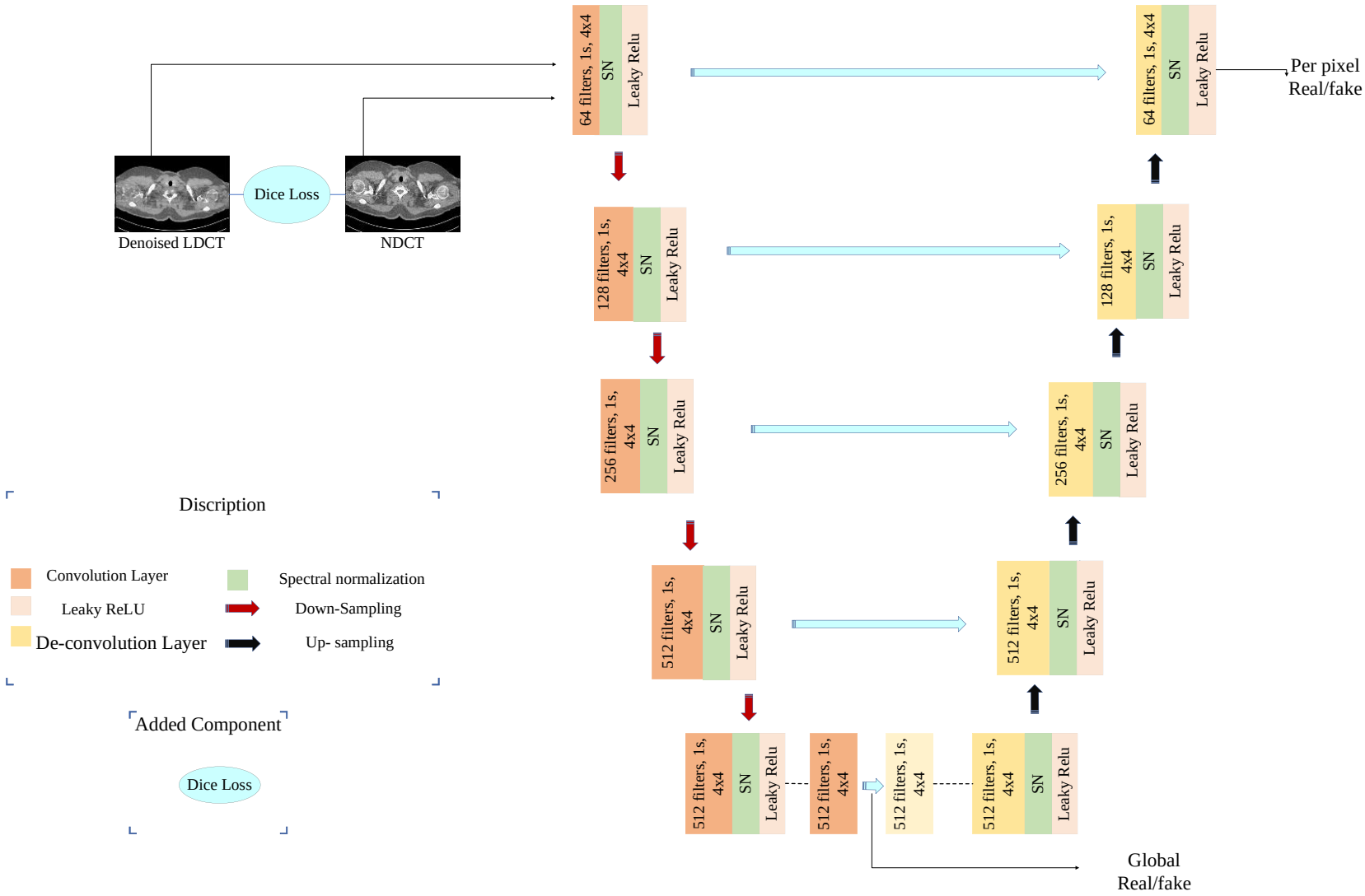



Figure 4.5 Discriminator Architecture

## 4.5. Model Learning Functions

The main objective of this work is to generate better-quality denoised LDCT. To achieve this objective different learning functions such as loss functions and attention mechanisms introduce to the network. The loss functions used in this work include adversarial loss which is defined in the context of least squares loss, reconstruction loss $l_1$ loss in the gradient domain, and instead of using MSE loss, which is what the baseline model does, we use dice loss for the pixel-wise loss.

### 4.5.1. Adversarial Loss

In this study, the sum of image and gradient branches of the discriminator is employed as adversarial loss using the context of Least Squares Loss to train the GAN framework rather than vanilla GAN. The objective function can be defined as follows:

$$L_{adv} = E\left[\underbrace{\left[D_{enc}^{img}(I_{den}) - 1\right]^2 + \left[D_{dec}^{img}(I_{den}) - 1\right]^2}_{\text{Image domain}}\right] + \underbrace{\left[D_{enc}^{grd}(\nabla(I_{den})) - 1\right]^2 + \left[D_{dec}^{grd}(\nabla(I_{den}) - 1\right]^2}_{\text{Gradient domain}} \tag{4.12}$$

Where ∇ denotes the Sobel operator to obtain the image gradient and the streak artifacts can be easily captured in this gradient domain as it can be illustrated in the figure 4.5.

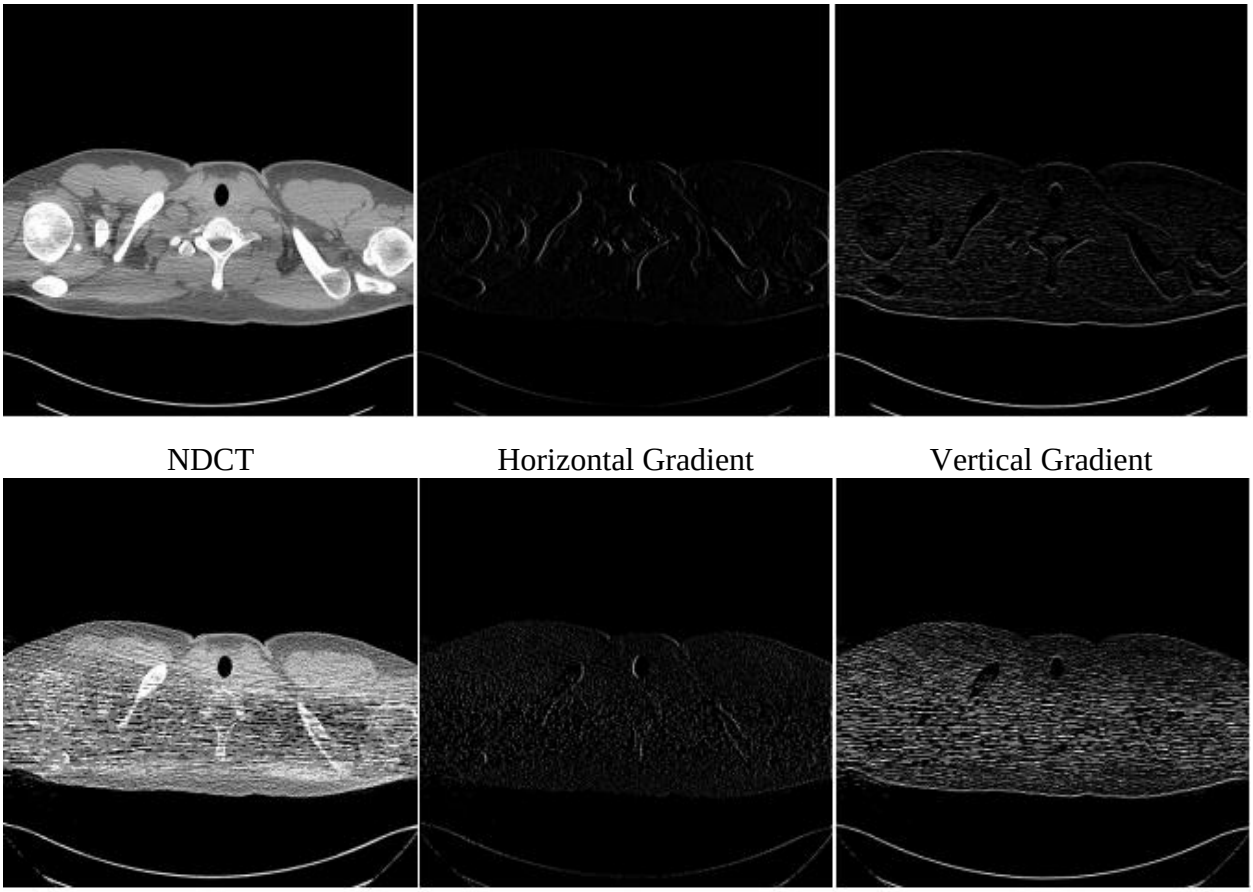



Figure 4.6 Vertical and Horizontal Gradients from a Pair of LDCT and NDCT Images.

Source: adapted from (Huang et al., 2022)

### 4.5.2. Reconstruction Loss

L1 is a common loss function that seeks to reduce the absolute differences between estimated and target values. The L1 loss function is more robust and unaffected by outliers in general. Following (Knyaz, Kniaz, & Remondino, 2019), l1 loss was used between the generated and real images in gradient domain discriminator .

$$L_{l1} = |It - \widetilde{It}| \tag{4.13}$$

Where $It$ is the ground truth image and $\widetilde{It}$ is the generated images

### 4.5.3. Dice Loss

The Dice coefficient is a popular metric in the computer vision domain for calculating image similarity and used for evaluating segmentation performance. Dice Coefficient has been adjusted to make it more tractable because it is non-convex in nature (Jadon, 2020). It was also implemented as a loss function known as Dice Loss later in 2016. As demonstrated in (Yue, 2021), dice loss is utilized to compute the task-oriented loss, allowing the denoising network to increase the specific features required by the discriminator network while also improving the overall efficiency of the medical image processing pipeline. Inspired by the above work, in this research dice loss is used as a pixel-wise loss Instead of MSE loss.

$$DL(y, \hat{p}) = 1 - \frac{2y\hat{p} + 1}{y + \hat{p} + 1} \tag{4.14}$$

Where y is the real image and $\hat{p}$ is fake or predicted image by the model, and 1 is added in numerator and denominator to ensure that the function is not undefined in edge case scenarios such as when y = $\hat{p}$ = 0.

## 4.6. Training Pseudo Code

The training algorithms employed in this work are described in this section. The efficient channel attention based LDCT image denoising by the GAN model was trained using the method in Table 4.1 in a similar way to how most GAN are trained. The goal of the entire configuration and training procedure is to lessen the noise in the LDCT image and the difference between the original image and the restored image.

**Algorithm: overall training algorithm of the proposed model**

**Inputs:** $I_{LD}$

**Outputs:** denoised LDCT $I_{den}$

**Ground Truth:** $I_{ND}$

For the number of training iteration do:

Train D:

$$\boldsymbol{L_{img}} = \mathbb{E}(\boldsymbol{I_{den}}, \boldsymbol{I_{ND}}) \left[\mathbf{1} - \frac{\mathbf{2}\, \boldsymbol{I_{ND}}(\boldsymbol{I_{den}}) + \mathbf{1}}{\boldsymbol{I_{ND}} + \boldsymbol{I_{den}} + \mathbf{1}}\right]$$

$$L_{grd} = \mathbb{E}(I_{den}, I_{ND}) \left|\nabla_{(I_{den})} - \nabla_{(I_{ND})}\right|$$

Update discriminator parameters to minimize classification loss.

Train G:

$$L_{adv\_loss} = \mathbb{E}\left[\left[D_{enc}^{img}(I_{den}) - 1\right]^2 + \left[D_{dec}^{img}(I_{den}) - 1\right]^2\right] + \left[D_{enc}^{grd}(\nabla(I_{den})) - 1\right]^2 + \left[D_{dec}^{grd}(\nabla(I_{den})) - 1\right]^2$$

$$L_{total} = \lambda_{adv\_loss} L_{adv\_loss} + \lambda_{img} \boldsymbol{L_{img}} + \lambda_{grd} L_{grd}$$

Update generator parameters to minimize reconstruction loss.

End for

The bold indicates the modified component

# CHAPTER FIVE

# 5. IMPLEMENTATION DETAILS

## 5.1. Chapter Overview

This chapter discusses the suggested model's implementation, as well as the working environment, data augmentation, ECA-DUGAN implementation, and the experiment class that was conducted.

## 5.2. Working Environment

This section discusses the environmental setup of hardware tools used for implementing the model.

Desktop:

- Processor: Intel®Intel™ i7-8700 CPU @ 20GHz 3.19 GHz
- Installed RAM: 8.00 GB
- Operating system: Window 10 pro
- System type: 64-bit operating system, x64-based processor

Desktop:

- Processor: Intel® Core™ i5-4590 CPU @ 3.30 GHz 3.30 GHz
- Installed RAM: 14.00 GB
- Operating system: Window 10 pro
- System type: 64-bit operating system, x64-based processor
- Graphics: Intel ® HD Graphics 4600
- GPU: GeForce RTX 2070 Super 6 GB RAM

## 5.3. Environmental Setup

- Anaconda: is an open-source tool that makes package administration easier. It is utilized to set up Python and its various modules. Anaconda version 1.9.7 is utilized to set up necessary libraries and modules.
- Jupyter notebook: an open-source, tool that integrates software code, explanatory text, computational output, and multimedia resources into a single page.

- Colab: is a free notebook that runs on the cloud and makes use of a free GPU. It lets you develop and run Python code and incorporates TensorFlow, OpenCV, Keras, and PyTorch.

### 5.4. Dataset Generation Implementation

Analyzing the effect of the patch size during training is crucial because of the discriminator's U-Net architecture. So that in the dataset generation implementation cropping the training images with patch size because training the denoising model from scratch with full image size i.e. 512 x 512, which is highly challenging. As a result, we trained our model using 64 x 64 images. (Huang et al., 2022) Shows small patch size can achieve better performance because the larger patch sizes may introduce training difficulties with less training samples.

```
patch_size = 64
def crop(ds):
    patches = []
    for left in range(0, ds.shape[0] - patch_size + 1, stride):
        for top in range(0, ds.shape[1] - patch_size + 1, stride):
            patches.append(ds[left: left + patch_size, top: top + patch_size])
    return patches
```

Sample code 5.1 Dataset generation implementation

### 5.5. Data Augmentation Implementation

As previously mentioned, this work uses the CutMix technique as a data augmentation strategy to generate the new training image by cutting patches from one image and pasting them into another.

```
def cutmix(mask_size):
    mask = torch.ones(mask_size)
    lam = np.random.beta(1., 1.)
    _, _, height, width = mask_size
    cx = np.random.uniform(0, width)
    cy = np.random.uniform(0, height)
    w = width * np.sqrt(1 - lam)
    h = height * np.sqrt(1 - lam)
    x0 = int(np.round(max(cx - w / 2, 0)))
    x1 = int(np.round(min(cx + w / 2, width)))
    y0 = int(np.round(max(cy - h / 2, 0)))
    y1 = int(np.round(min(cy + h / 2, height)))
    mask[:, :, y0:y1, x0:x1] = 0
    return mask
```

Sample code 5.2 Data augmentation implementation

### 5.5. ECA-DUGAN for the denoising of LDCT

One generator and two discriminator networks are included in the proposed model. The generator network is a REDCNN generator with encoders, a residual block, and a decoder, as

discussed in section 4.3. Both encoders and decoders are 10-layer neural networks with 32 filters for computational efficiency, followed by a ReLU activation function. There are ten residual skip connections as (Huang et al., 2022). The layers of the generator and discriminator architecture are described in Tables 5.1 and 5.2.

```
class Generator(nn.Module):
    def __init__(self, in_channels=1, out_channels=96, num_layers=10, kernel_size=3, padding=1):
        super(Generator, self).__init__()
        encoder = [
            nn.Conv2d(in_channels, out_channels, kernel_size=kernel_size, stride=1, padding=padding),

        ]
        instnorm = [
            nn.InstanceNorm2d(out_channels,  affine=True)

        ]
        # channel = [
        #     ChannelAttention(out_channels, ratio=16)
        # ]

        decoder = [
            nn.ConvTranspose2d(out_channels, in_channels, kernel_size=kernel_size, stride=1, padding=padding)
        ]
        for _ in range(num_layers):
            encoder.append(
                nn.Conv2d(out_channels, out_channels, kernel_size=kernel_size, stride=1, padding=padding),

            )
            instnorm.append(
                nn.InstanceNorm2d(out_channels, affine=True)
            )
```

```
            decoder.append(
                nn.ConvTranspose2d(out_channels, out_channels, kernel_size=kernel_size, stride=1, padding=padding),

            )


        self.eca = eca_layer(out_channels, kernel_size)
        self.encoder = nn.ModuleList(encoder)
        self.instnorm = nn.ModuleList(instnorm)
        self.decoder = nn.ModuleList(decoder)

        self.__init_weights()
```

```
def forward(self, x: torch.Tensor):
    x = x + self.eca(x)
    residuals = []
    for block in self.encoder:
        residuals.append(x)
        x = F.relu(block(x), inplace=True)
    for inst in self.instnorm:
        x = F.relu(inst(x), inplace=True)
    for residual, block in zip(residuals[::-1], self.decoder[::-1]):
        x = F.relu(block(x) + residual, inplace=True)
    return x
```

Sample code 5.3 Generator function implementation

Then the least squared loss is computed to assess whether the generated image is real or fake. The detailed implementation is discussed in appendix C.

Table 5.1 Content of the generator architecture

| Generator | |
|---|---|
| Layers in the encoder | Content of each block |
| 1 | Convolution - (filters = 32, kernel = 3 x 3, stride = 1, padding = 1) |
| 2-10 | Convolution - (filters = 32, kernel = 3 x 3, stride = 1, padding = 1)<br>instance normalization, Relu |
| Layers in the decoder | |
| 1-10 | Deconvolution – (filters = 32, kernel = 3 x 3, stride = 1, padding = 1) , ReLu |

Table 5.2 Content of the discriminator architecture

| Discriminator | |
|---|---|
| Layers in the encoder | Content of each block |
| 1 | Convolution - (filters = 64, kernel = 4 x 4, stride = 1, padding = 1), Spectral normalization, Leaky ReLu |
| 2 | Convolution - (filters = 128, kernel = 4 x 4, stride = 1, padding = 1), Spectral normalization, Leaky ReLu |
| 3 | Convolution - (filters = 256, kernel = 4 x 4, stride = 1, padding = 1), Spectral normalization, Leaky ReLu |
| 4-5 | Convolution - (filters = 512, kernel = 4 x 4, stride = 1, padding = 1), Spectral normalization, Leaky ReLu |
| 6 | Convolution - (filters = 512, kernel = 4 x 4, stride = 1, padding = 1) |
| Layers in the decoder | |
| 1 | Convolution - (filters = 64, kernel = 4 x 4, stride = 1, padding = 1), Spectral normalization, Leaky ReLu |
| 2 | Convolution - (filters = 128, kernel = 4 x 4, stride = 1, padding = 1), Spectral normalization, Leaky ReLu |
| 3 | Convolution - (filters = 256, kernel = 4 x 4, stride = 1, padding = 1), Spectral normalization, Leaky ReLu |
| 4-5 | Convolution - (filters = 512, kernel = 4 x 4, stride = 1, padding = 1), Spectral normalization, Leaky ReLu |
| 6 | Convolution - (filters = 512, kernel = 4 x 4, stride = 1, padding = 1) |

### 5.5.1. Efficient Channel Attention Implementation

An efficient channel-attention layer, as mentioned in section 4.3.2, assists the generator network in producing images with fine details in every location that are precisely coordinated with fine characteristics in distant sections of the image. Before proceeding to the generator's first encoder layer, the outputs of efficient channel-attention layer is concatenated with the input LDCT images.

```
class eca_layer(nn.Module):

    def __init__(self, channel, k_size):
        super(eca_layer, self).__init__()
        self.avg_pool = nn.AdaptiveAvgPool2d(1)
        self.conv = nn.Conv1d(1, 1, kernel_size=k_size, padding=(k_size - 1) // 2, bias=False)
        self.sigmoid = nn.Sigmoid()

    def forward(self, x):
        # feature descriptor on the global spatial information
        y = self.avg_pool(x)

        # Two different branches of ECA module
        y = self.conv(y.squeeze(-1).transpose(-1, -2)).transpose(-1, -2).unsqueeze(-1)

        # Multi-scale information fusion
        y = self.sigmoid(y)

        return x * y.expand_as(x)
```

Sample code 5.4 Efficient Channel attention implementation

### 5.5.2. Dice Loss Implementation

As described in section 4.5.3 dice loss is utilized to compute the task-oriented loss, allowing the denoising network to increase the specific features required by the discriminator network while also improving the overall efficiency of the medical image processing pipeline.

```
class DiceLoss(nn.Module):

    def __init__(self, weight=None, ignore_index=[], **kwargs):
        super(DiceLoss, self).__init__()
        self.kwargs = kwargs
        if weight is not None:
            self.weight = weight / weight.sum()
        else:
            self.weight = None
        self.ignore_index = ignore_index

    def forward(self, predict, target):
        assert predict.shape == target.shape, 'predict & target shape do not match'
        dice = BinaryDiceLoss(**self.kwargs)
        total_loss = 0
        total_loss_num = 0

        for c in range(target.shape[1]):
            if c not in self.ignore_index:
                dice_loss = dice(predict[:, c], target[:, c])
                if self.weight is not None:
                    assert self.weight.shape[0] == target.shape[1], \
                        'Expect weight shape [{}], get[{}]'.format(target.shape[1], self.weight.shape[0])
                    dice_loss *= self.weight[c]
                total_loss += dice_loss
                total_loss_num += 1
```

Sample code 5.5 Dice Loss Implementation

## 5.6. Discriminator Network Implementation

The discriminator used in this work is U-Net based discriminator which has a lot of advantages over other traditional discriminator architectures such the patch discriminator, pixel discriminator, (Knyaz et al., 2019), and traditional global discriminator. The U-Net based discriminator has the benefits of both worlds producing higher quantitative findings and denoising quality as opposed to pixel discriminators that only capture per-pixel differences and conventional classification discriminators that solely concentrate on global structure. There are two independent discriminators in both image and gradient domains, each of which follows a U-Net architecture. Specifically, $D_{enc}$ has 6 down-sampling ResBlocks (Kaiming He, Xiangyu Zhang, Shaoqing Ren, 2016) with increasing number of filters; i.e. 64, 128, 256, 512, 512, and 512. At the bottom of $D_{enc}$, a fully-connected layer is used to output the global confidence score. Similarly, $D_{dec}$ used the same number of ResBlocks in a reverse order to process the bilinearly up-sampled features and the skip residuals of the same resolution, followed by a 1×1 convolutional layer to output the per-pixel confidence map. Most importantly, a spectral normalization layer (Miyato et al., 2018)and a Leaky ReLU activation with a slope of 0.2 for negative input follow each convolutional layer of D except the last one.

```
class UNet(nn.Module):
    def __init__(self, repeat_num, use_tanh=False, use_sigmoid=False, skip_connection=True, use_discriminator=True,
                 conv_dim=64, in_channels=1):
        super().__init__()
        self.use_tanh = use_tanh
        self.skip_connection = skip_connection
        self.use_discriminator = skip_connection
        self.use_sigmoid = use_sigmoid

        filters = [in_channels] + [min(conv_dim * (2 ** i), 512) for i in range(repeat_num + 1)]
        filters[-1] = filters[-2]

        channel_in_out = list(zip(filters[:-1], filters[1:]))

        self.down_blocks = nn.ModuleList()

        for i, (in_channel, out_channel) in enumerate(channel_in_out):
            self.down_blocks.append(DownBlock(in_channel, out_channel, downsample=(i != (len(channel_in_out) - 1))))

        last_channel = filters[-1]
        if use_discriminator:
            self.to_logit = nn.Sequential(
                nn.LeakyReLU(negative_slope=0.2, inplace=True),
                nn.AdaptiveAvgPool2d(output_size=1),
                nn.Flatten(),
                nn.Linear(last_channel, 1)
            )
```

```
        self.up_blocks = nn.ModuleList(list(map(lambda c: UpBlock(c[1] * 2, c[0]), channel_in_out[:-1][::-1])))
        #self.Att5 = Attention_block()

        self.conv = double_conv(last_channel, last_channel)
        self.conv_out = nn.Conv2d(in_channels, 1, 1)
        self.__init_weights()
```

Sample code 5.6 Discriminator Implementation

## 5.7. Hyperparameter Configuration

An established parameter, such as batch size, number of epochs, optimization, learning rate, and loss weight, is referred to as a hyper-parameter. These variables are modifiable and directly affect how well a model trains. Adam optimizer with fixed learning-rate = $10^{-4}$ was used to update the network weight. Optimizers are methodological approaches that are used to modify the characteristics of the neural network, such as weights and learning rate, to minimize losses. Adam is said to combine the advantages of two different stochastic gradient descent algorithms. Particularly AdaGrad and RMSProp (Kingma & Ba, 2015), it is simple to implement, computationally effective, and memory-light. The amount by which the weights are altered during exercise is known as the step size, also referred to as the "learning rate." The weight loss and learning rate employed in this study are comparable to those in (Huang et al., 2022). The model is trained over 10000 iterations with a batch size of 1.

Table 5.3 Hyperparameter Configuration

| Hyperparameter | Values |
|---|---|

| | |
|---|---|
| Batch size | 1 |
| Number of iteration | 10000 |
| Optimizer | Adam optimizer |
| Learning Rate | $10^{-4}$ |
| Loss weight for adversarial loss ($\lambda_{adv\_loss}$) | 0.1 |
| Loss weight for dice loss in image domain discriminator ($\lambda_{img}$) | 1 |
| Loss weight for L1 in gradient domain discriminator ($\lambda_{grd}$) | 20 |

## 5.8. Experiment Class

It is required to conduct many experiments and test them in order to evaluate the essence of adopting attention mechanisms and better pixel wise loss in the improvement of denoised LDCT image quality. There are six separate experiments carried out as shown in table 5.4. The implementation of the baseline work constitutes the first experiment. As stated in section 3.7, DUGAN used dual discriminators to identify both global and local differences between images that had been denoised and those that had received a normal dose. The image domain discriminator provided per-pixel feedback to the denoising network through the outputs of the Unet focused on the global structure at the semantic level through the middle layer of the UNet, while the gradient domain discriminator alleviate the artifacts caused by photon starvation and enhance the edge of the denoised LDCT images.

Initial work of the proposed model is the second experiment. In this experiment, a channel attention block was incorporated in the last layer of generator encoder network along with the addition of instance normalization in the generator encoder network while keeping the pixel wise loss the same as in the baseline work, which is MSE loss. In the third and fourth trial, an attempt was made to determine the pixel-wise loss that could handle the denoising process, and as a consequence, dice loss rather than MSE loss produced results with less blur. And in the remaining experiments dice loss was used as a pixel wise loss. In the fifth experiment, additional research was conducted to identify more effective attention mechanisms. To this end, effective channel attention block and spatial attention techniques were applied in various generator network regions. In order to retrieve the finer structural details of the denoised LDCT image, the

model was finally trained using only the efficient channel attention block, as described in section 4.3.2. The noise reduction and structural detail preservation are guided by the attention map M, which is pumped into the generator after concatenated with the input LDCT images channel-wisely. The overall proposed model is represented by this experiment.

Table 5.4 List of experiment class

| Notation | Experiment | Dataset |
|---|---|---|
| DU-GAN | Baseline | NIHAAPM-Mayo Clinic Low-Dose CT with 10% Normal dose chest CT |
| CA + IN+DU-GAN | With the addition of Channel Attention block in the generator-encoder and instance normalization while keeping the pixel wise loss the same as in the baseline work, which is MSE loss | NIHAAPM-Mayo Clinic Low-Dose CT with 10% Normal dose chest CT |
| CA + IN+ DU-GAN - (dice loss + MSE loss) | With combination of dice loss and MSE loss as the pixel wise loss instead of MSE loss | NIHAAPM-Mayo Clinic Low-Dose CT with 10% Normal dose chest CT |
| CA + IN+DU-GAN -(dice loss) | With dice loss as the pixel wise loss instead of MSE loss | NIHAAPM-Mayo Clinic Low-Dose CT with 10% Normal dose chest CT |
| ECA + IN+ SA+ DU-GAN-(dice loss) | With the addition of both Efficient channel attention block and spatial attention layer to the generator network | NIHAAPM-Mayo Clinic Low-Dose CT with 10% Normal dose chest CT |
| ECA + IN+DU-GAN-(dice loss) | With only the addition of Efficient channel attention block | NIHAAPM-Mayo Clinic Low-Dose CT with 10% Normal dose chest CT |

# CHAPTER SIX

# 6. RESULTS AND DISCUSSION

## 6.1. Chapter Overview

This chapter discusses the results of baseline research, the various experiments undertaken, including the proposed approach, and compares the results with figure, graphs, and tables.

## 6.2. Result of the Study

### 6.2.1. Experimental Results

In order to obtain better findings for this study, six experiments were conducted, and all of them used the same dataset, hardware, and software configuration to allow for comparison. The findings for the NIHAAPM-Mayo Clinic Low-Dose CT dataset are displayed.

#### 6.2.2.1. DU-GAN

DU-GAN contains one generator and two discriminators. The first discriminator is in the image domain branch and used to penalize the generator generating photo-realistic denoised LDCT while the second discriminator in the gradient domain branch encourages better edge while alleviating the streak artifacts caused by photon starvation. (Huang et al., 2022) Additionally, the discriminator in each branch employs a U-Net based architecture to encourage the generator focusing both global structure and local details.

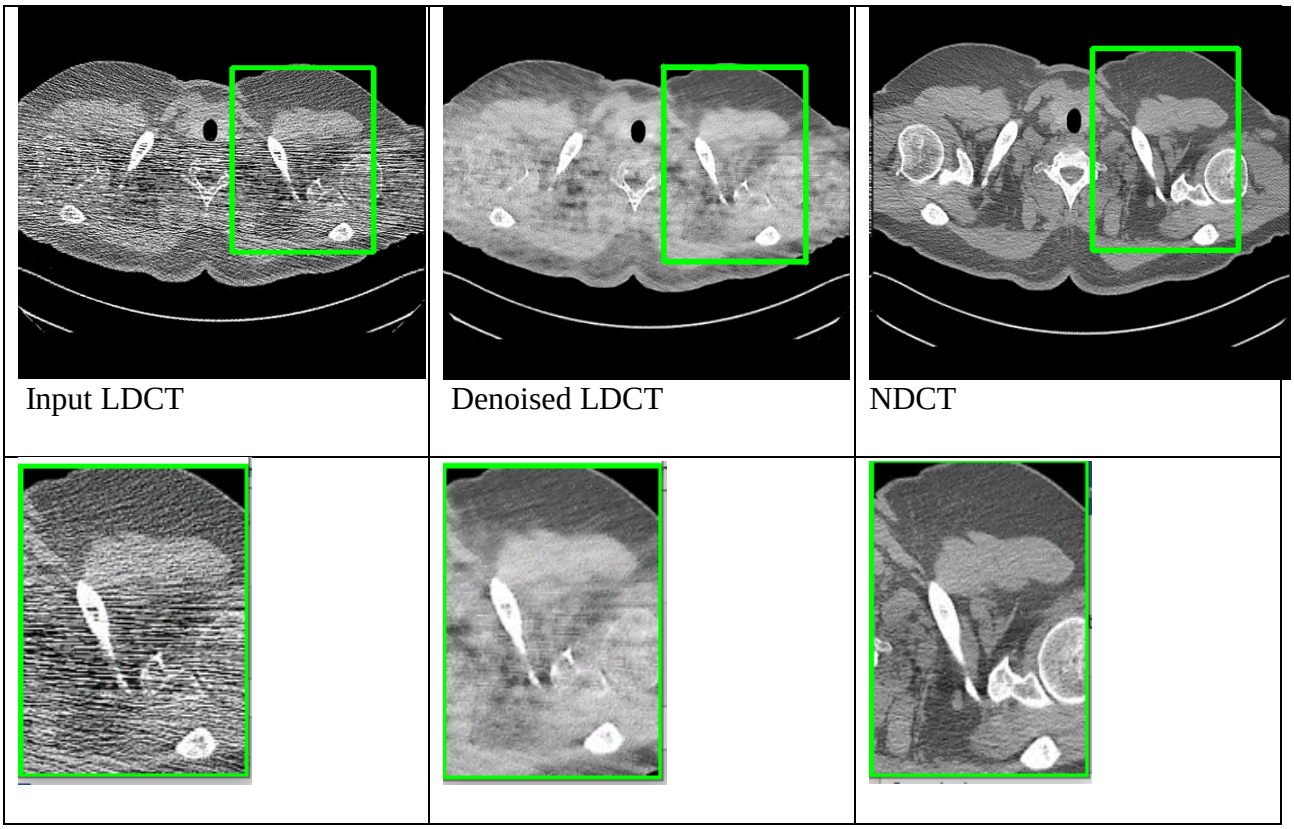



Figure 6.1 Results of DUGAN

In the first and second column there are input LDCT and NDCT images and the second column shows results of DU-GAN respectively.

As seen in figure 6.1, there is a lot of blur and fog in the second column of the denoised LDCT image, which affects the structural detail of the CT scan image. This demonstrates that there is a probability that the DU-GAN network may fail to preserve structural information due to lack of attention network. Furthermore, there is also a quality problem due to the blur caused by using MSE loss as pixel wise loss.

#### 6.2.2.2. CA + IN+DU-GAN

To solve the problem of preserving the structural detail channel attention layer was added at the decoder of the generator along with instance normalization in the generator network while keeping the pixel-wise loss as it is in the baseline work.

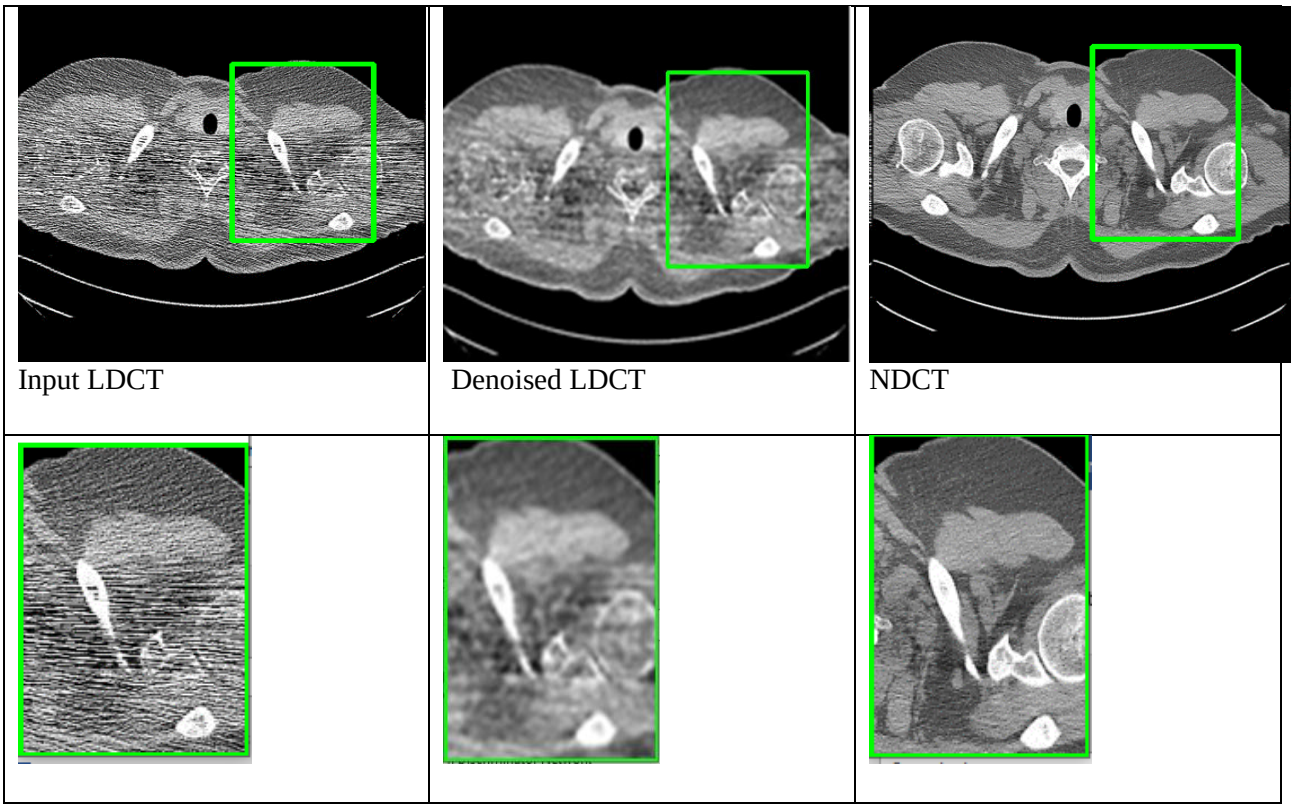



Figure 6.2 Results of CA + IN+ DU-GAN

In the first and third column there are input LDCT and NDCT images and in the second column shows results of CA+IN+DU-GAN respectively.

As shown in figure 6.2, the blurry decreases compared to the previous experiment but there are still noise artifacts at the generated image. Moreover, the structural details are still not clear.

**6.2.2.3. CA + IN+ DU-GAN - (dice loss + MSE loss)**

The second experiment was conducted to address the blurriness issue, which was mostly brought on by the MSE loss in the image domain discriminator.

While maintaining the other elements from the prior experiment, dice loss is introduced and used in conjunction with MSE loss in this experiment to examine its effects.

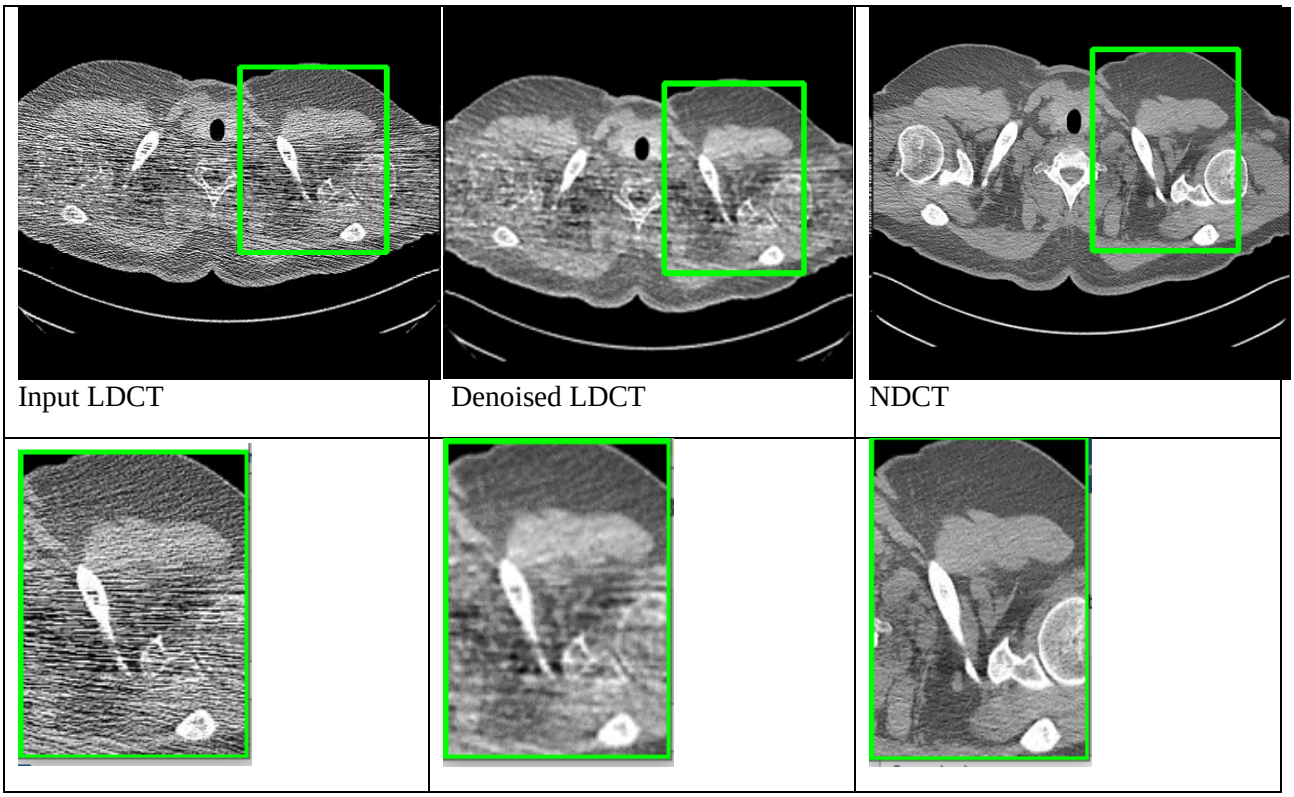



Figure 6.3 Results of CA + IN+ DU-GAN + (dice loss + MSE loss)

As seen in the figure 6.3 this model was unable to significantly reduce blurriness visually, but as can be seen in section 6.5, the quantitative results of the evaluation metrics show modest improvement.

#### 6.2.2.4. CA + IN+DU-GAN - (dice loss)

To further examine the effect of dice loss, in this experiment only dice loss is used as a pixel wise in the image domain discriminator.

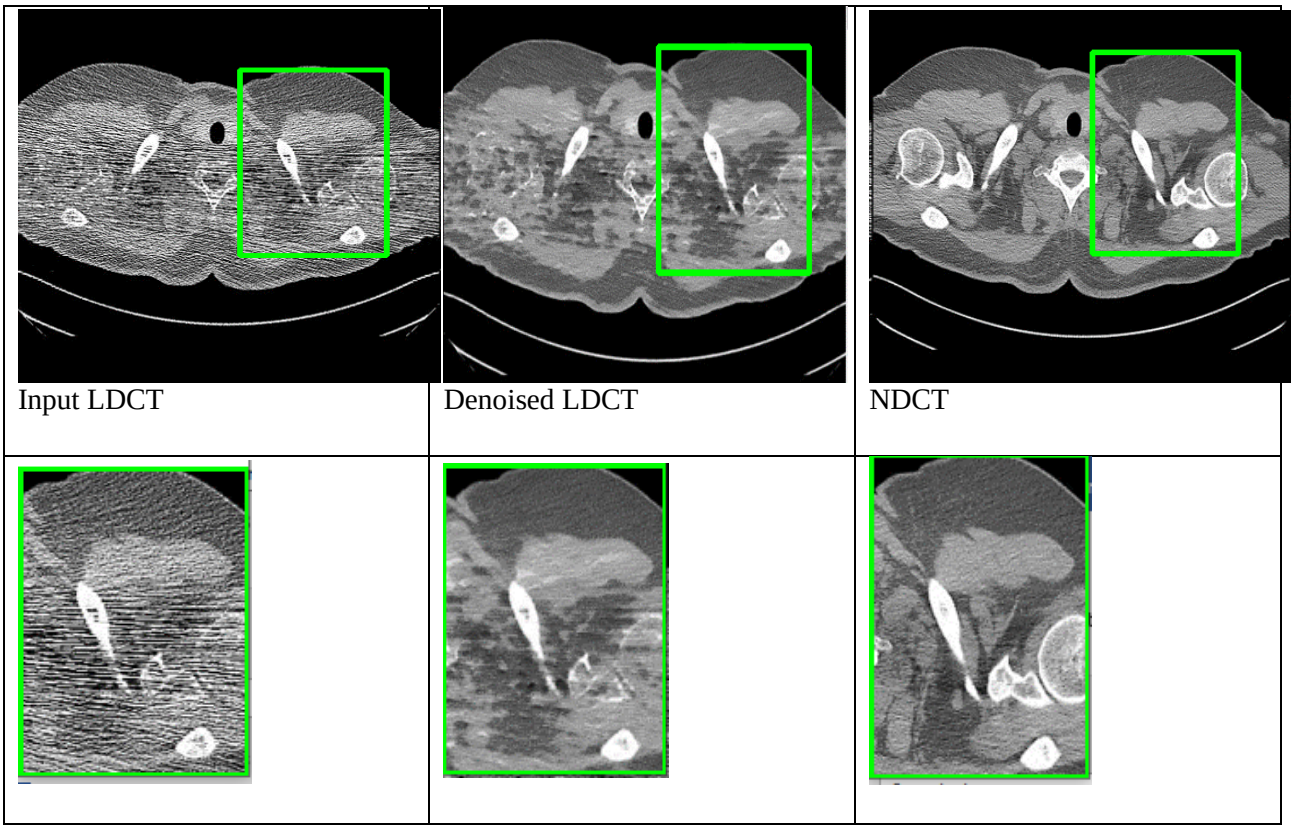



Figure 6.4 Results of CA + IN+ DU-GAN + Dice Loss

As can be seen in the figure 6.4, the addition of dice loss greatly reduces blurriness visually. As can be shown in section 6.5, the quantitative results of the assessment metrics also improve when compared to the previous one. However, compared to the NDCT image, the produced images lack some structural detail.

#### 6.2.2.5. ECA + IN+ SA+ DU-GAN – (dice loss)

To solve the lack of structural detail issue in the previous experiment, in this experiment Efficient channel attention block is added before the generator network, spatial attention block is added between encoder and decoder in the generator network while keeping the dice loss as per pixel loss in image discriminator branch.

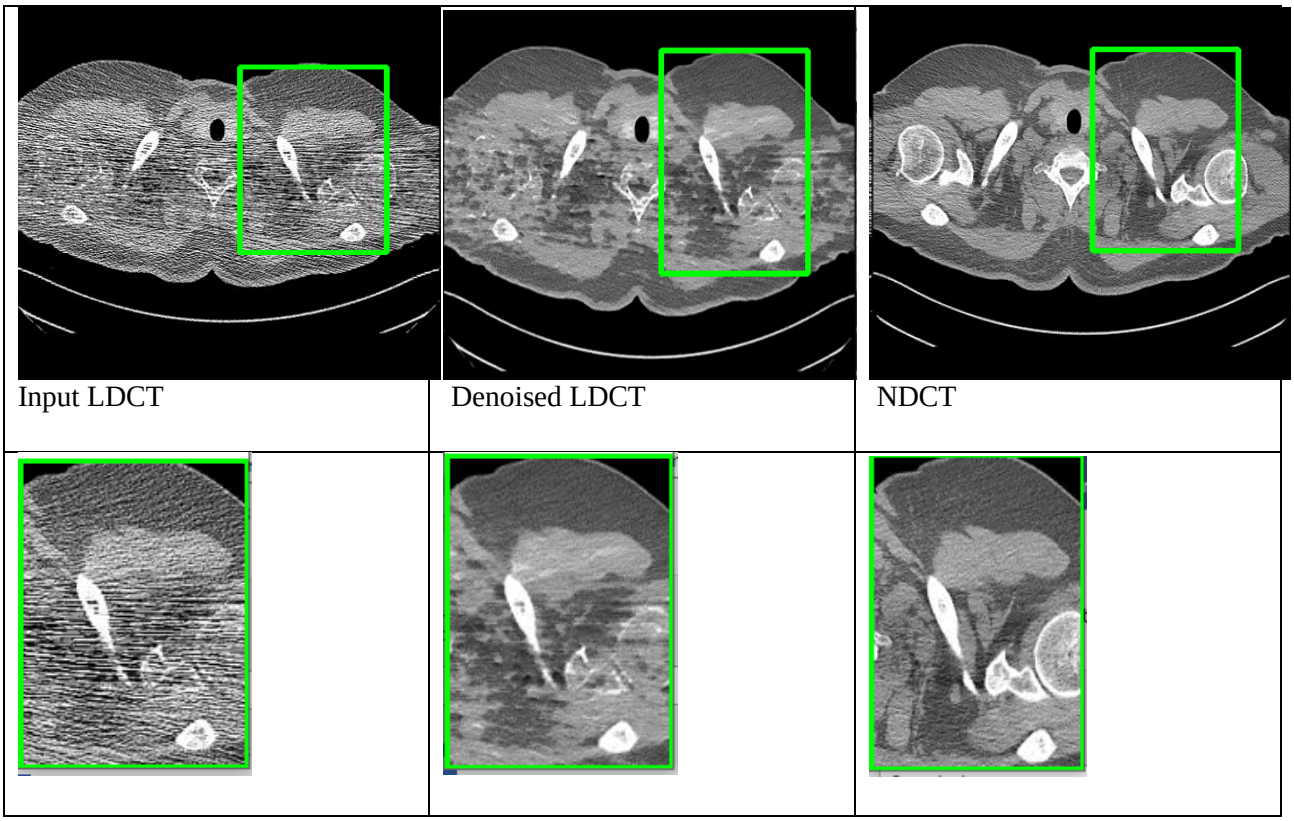



Figure 6.5 Results of ECA + IN+ SA+ DU-GAN+ Dice Loss

As shown in the figure 6.5 even though there was a slight enhancement according to the evaluation metrics as it is mentioned in section 6.4.1., the result images do not show that much improvement in visual quality when compared to the previous experiment. This leads us to examine the next experiment.

#### 6.2.2.6. ECA + IN+DU-GAN – (dice loss)

In this experiment, the network is trained without spatial attention block in the generator network to show the effect of efficient channel attention block has on the network.

As shown in figure 6.6 below, the denoised LDCT image has good quality for the dataset compared to the previous experiments. The result shows that the efficient channel attention block on its own can give the required cross channel communication which brings a clear improvement to the model to generate better quality denoised LDCT images.

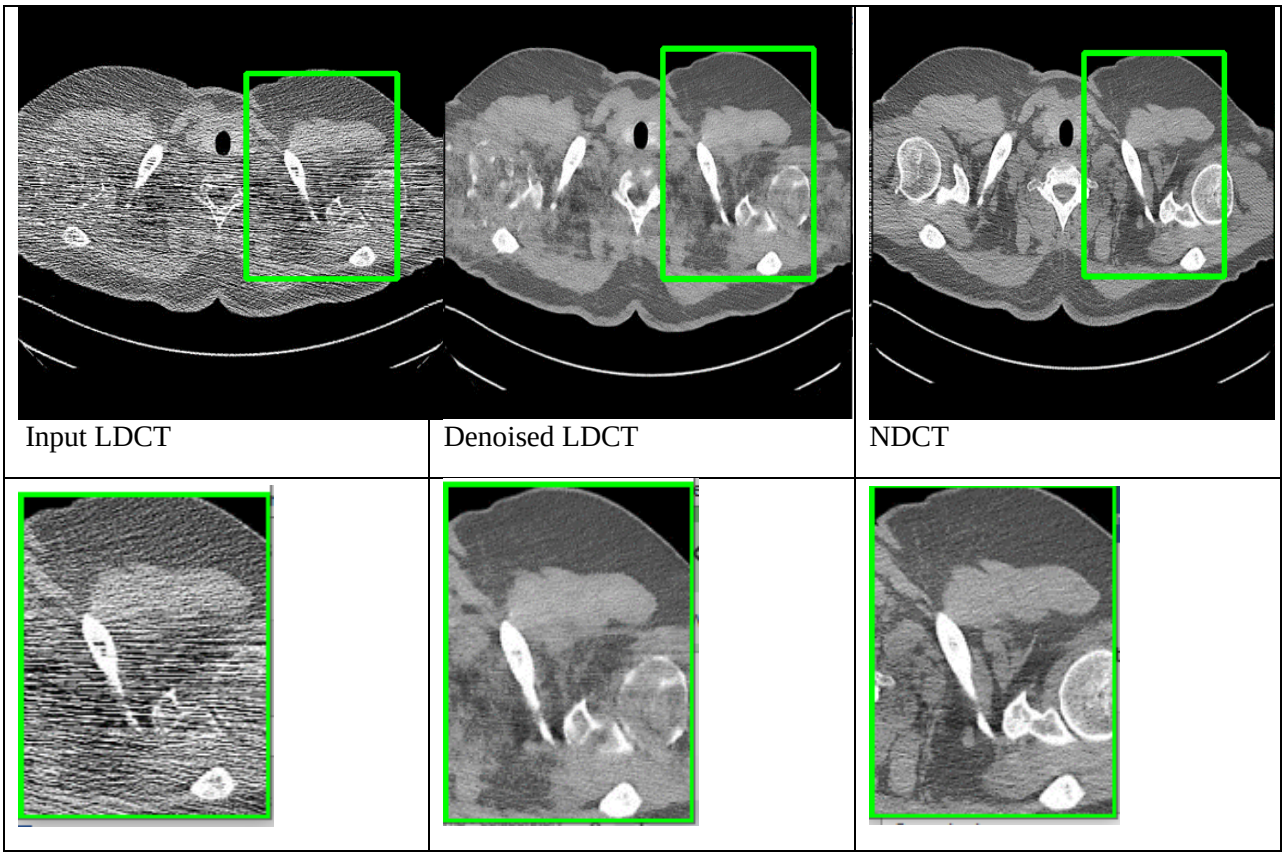



Figure 6.6 Results of ECA + IN + DU-GAN+ Dice Loss

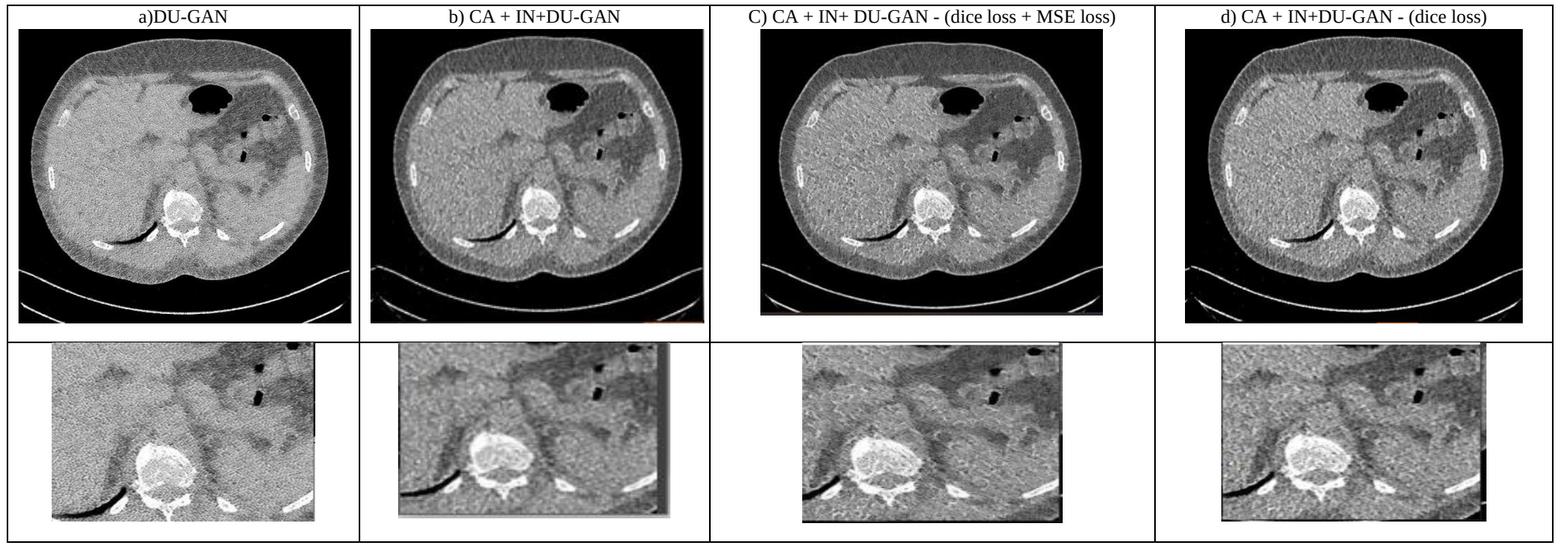



Figure 6.7 Transverse CT images from the Mayo-10%

## 6.3. Sample Training Log for ECA-DUGAN

Figure 6.8 shows the generator loss of proposed model, at the first number of iterations, the loss was around 16.8, and even if there is an up and down in the loss value, the loss decreases smoothly throughout the training epochs. At number of iteration 10000 the loss is around 2.73.

Figure 6.9 shows the discriminator loss of proposed model, at the first number of iteration, the loss was around 6.48. There is an up and down in the loss value. However it decreases throughout the final number of iterations. At iteration number 10000 the loss is around 0.85783756.

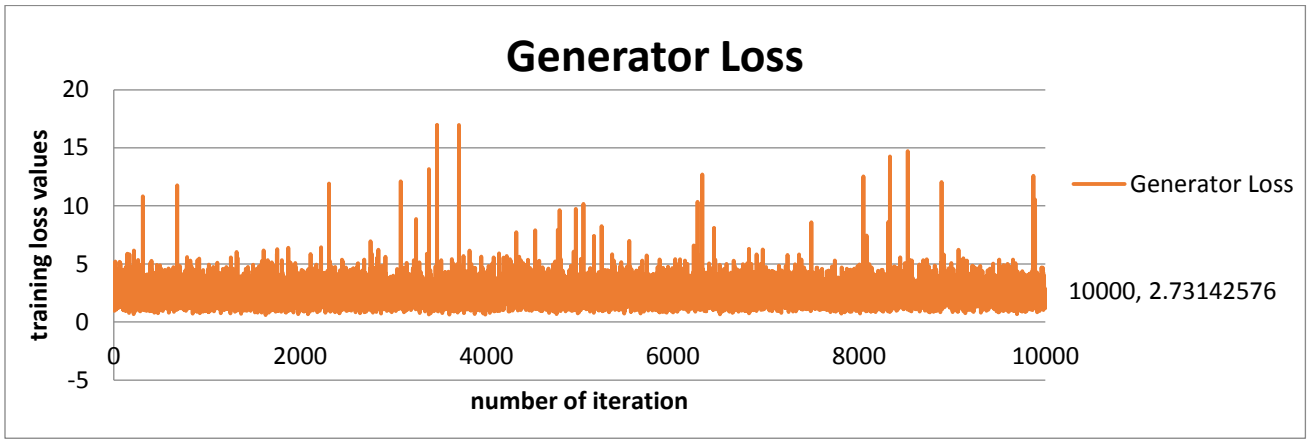



Figure 6.8 Generator Loss Graph

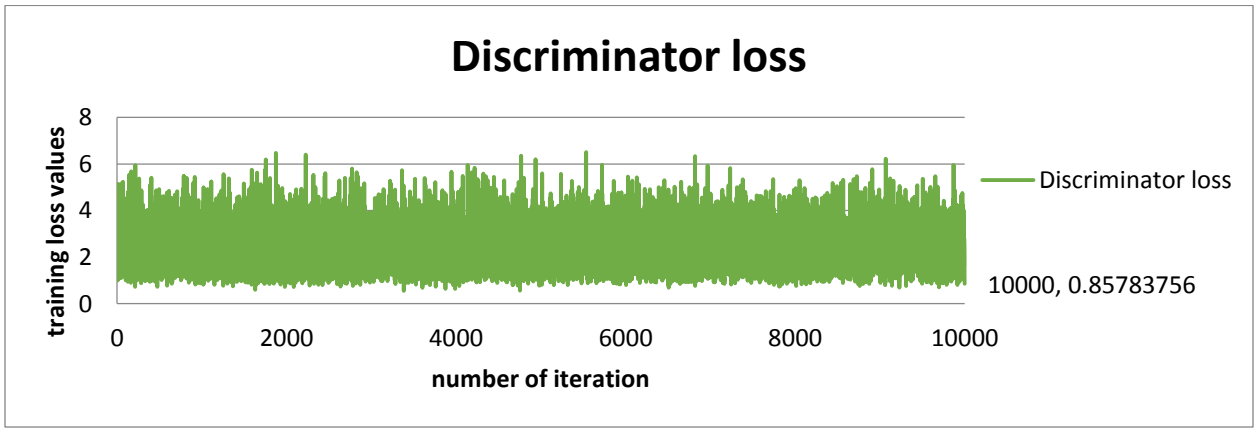



Figure 6.9 Discriminator Loss Graph

## 6.4. Evaluation Metrics

### 6.4.1. Peak Signal to Noise Ratio, Structural Similarity Index Measure and Root Mean Squared Error

Three objective evaluation metrics (PSNR, SSIM, and RMSE) were utilized to assess the execution of the model. Table 6.1 summarizes the result of the learned model to show their reconstruction ability via PSNR, SSIM, and RMSE metrics to measure the similarity of restored images and ground truth image (in this case NDCT image). Those metrics are calculated by using 6439 test LDCT images from the NIHAAPM-Mayo Clinic Low-Dose CT dataset. Higher PSNR and SSIM values and lower RMSE values indicate that the generated image has become close to the original image.

Table 6.1 PSNR, SSIM, and RMSE for all experiments

| Number | Experiments | SSIM | PSNR | RMSE |
|---|---|---|---|---|
| 1 | DUGAN | 0.72109 | 20.40010 | 0.09821 |
| 2 | CA + IN+DU-GAN | 0.73504 | 21.32325 | 0.08902 |
| 3 | CA + IN+DU-GAN + dice loss + MSE loss | 0.73067 | 21.76348 | 0.08429 |
| 4 | CA + IN+DU-GAN + dice loss | 0.73096 | 21.81198 | 0.08400 |
| 5 | ECA + IN+SA+ DU-GAN + dice loss | 0.73258 | 21.92848 | 0.08286 |
| 6 | ECA + IN+DU-GAN + dice loss | **0.73830** | **22.04259** | **0.08170** |

## 6.5. Results Discussion

### 6.5.1. Discussion on SSIM Result

The SSIM evaluates the perceived quality of images and visual similarity as well as structural similarity between NDCT images and denoised LDCT images. As demonstrated in Table 6.1 the CA + IN+DU-GAN model scores 0.73504 which is better than the baseline work. The CA + IN+DU-GAN + dice loss + MSE loss, CA + IN+DU-GAN + dice loss, and ECA + IN+SA+ DU-GAN+ dice loss score 0.73067, 0.73096 and 0.73258 respectively. The ECA + IN+DU-GAN+ dice loss improved the SSIM score from 0.73258 to 0.73830 indicating that the dice loss, Efficient channel attention along with instance normalization improve the quality of the generated denoised LDCT images.

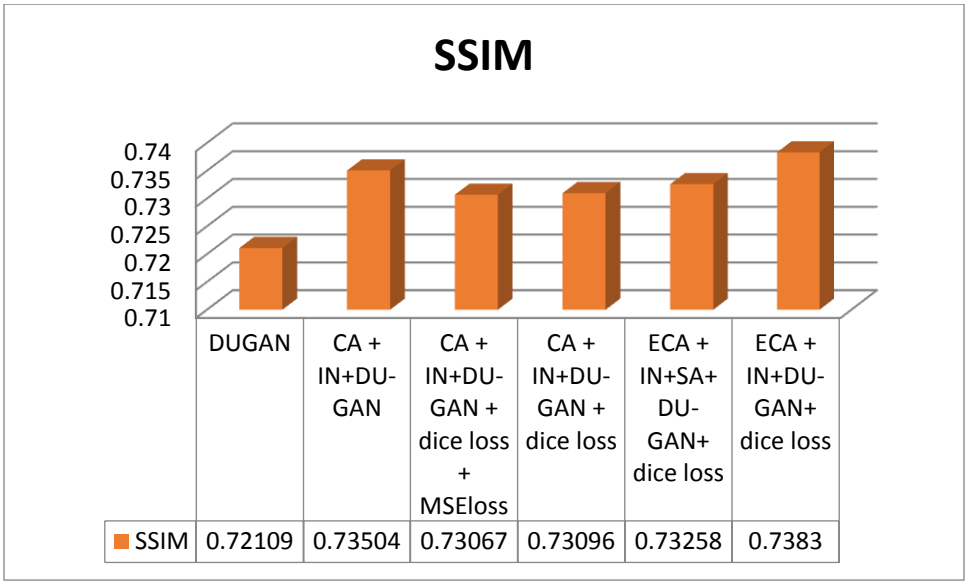



Figure 6.10 SSIM Score of the Six Models

The y-axis indicates the SSIM score and the x-axis indicates the six experiments

### 6.5.2. Discussion on PSNR result

PSNR is another metric used for evaluating the quality of the generated image in pixel space. As demonstrated in Table 6.1 the DU-GAN scores the lowest PSNR which is 20.40010. The CA + IN+DU-GAN scores 21.32325, indicating that the quality of the produced LDCT image is higher than the prior experiment. CA + IN+DU-GAN + dice loss + MSE loss , CA + IN+DU-GAN + dice loss, and ECA + IN+SA+ DU-GAN+ dice loss score 21.76348, 21.81198, and 21.92848 respectively all of them are a bit higher than CA + DU-GAN since they generate image

sequences somehow sharp than the CA + IN+DU-GAN. The ECA + IN+DU-GAN+ dice loss model scores 22.04259 which is the highest of all the other experiments, indicating that the quality of the image sequences generated by this model is better than previous experiments.

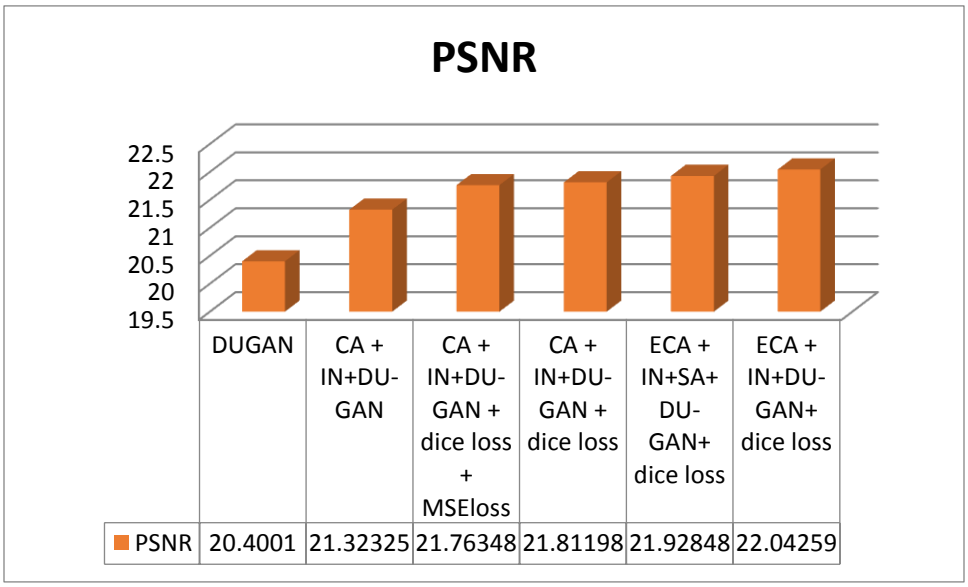



Figure 6.11 PSNR Score of the Six Models

The y-axis indicates the PSNR score and the x-axis indicates the six experiments

### 6.5.3. Discussion on RMSE result

RMSE is also another metric used in this work for evaluating the quality of the generated image in pixel space like PSNR. As demonstrated in Table 6.1 the DU-GAN scores the highest RMSE which is 0.09821. The CA + IN+DU-GAN scores 0.08902, indicating that the quality of the produced LDCT image is higher than the prior experiment. CA + IN+DU-GAN + dice loss + MSE loss, CA + IN+DU-GAN + dice loss , and ECA + IN+SA+ DU-GAN+ dice loss score 0.08429, 0.08400 and 0.08286 respectively all of them are a bit higher than CA + IN+DU-GAN since they generate image sequences somehow sharp than the CA + IN+DU-GAN. The ECA + IN+DU-GAN+ dice loss model scores 0.08170 which is the lowest of all the other experiments, indicating that the quality of the image sequences generated by this model is better than previous experiments.

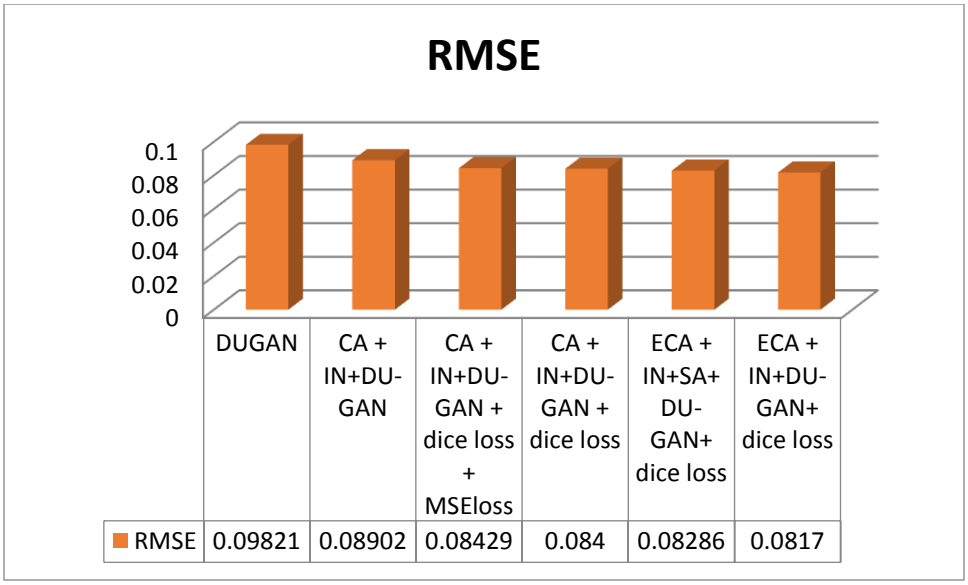



Figure 6.12 RMSE Score of the Six Models

The y-axis indicates the RMSE score and the x-axis indicates the six experiments

### 6.5.4. Research Question Discussion

Every research question posed in the first chapter is addressed by this study. The following is how this study responds to each question:

Answer for Q1: The better learning constraints were identified for enhancing the denoised LDCT image quality such as efficient channel-wise attention was identified for improving the structural details of denoised LDCT image and dice loss was identified for compute the task-oriented loss between the denoised LDCT image and NDCT, allowing the denoising network to increase the specific features required by the discriminator network. Instance normalization was also identified for faster convergence and to improve training effectiveness of the denoising network.

Answer for Q2: Efficient channel attention was added in the generator network as an adequate cross-channel communication block, the experimental result shows a better result regarding the denoised LDCT image quality. The result of the SSIM, PSNR and RMSE shows the effectiveness of this attention for enhancing the quality of the denoised LDCT image by improving the SSIM score from 0.73258 to 0.73830, the PSNR score from 21.92848 to 22.04259 and the RMSE score from 0.08286 to 0.08170. To show the effect of another attention mechanisms, which can provide cross channel communication block, separate experiments were

conducted so based on the evaluation metrices as shown in table 6.1, the channel attention improved the result of baseline model from 0.72109 to 0.73504, 20.40010 to 21.32325, 0.09821 to 0.08902 based on the SSIM, PSNR, and RMSE results respectively. The efficient channel attention and spatial attention improves the results of the channel attention from 0.73096 to 0.73258, 21.81198 to 21.92848, 0.08400 to 0.08286 based on the SSIM, PSNR, and RMSE results respectively. From this answer, this study finds that the identified attention mechanism so called efficient channel attention have a great influence on quality enhancement.

Answer for Q3: Dice loss was added to the image discriminator model to saw the effect of this loss, the experimental result shows a better result regarding enhancing the structural detail similarity between the generated denoised LDCT image and the target NDCT image. Based on the evaluation metrices as shown in table 6.1, the dice loss improves the SSIM, PSNR, and RMSE result from 0.72109 to 0.73096, 20.40010 to 21.81198 and 0.09821 to 0.08400. From this answer, this study generalized that the identified dice loss shows, dice loss has a positive impact on the reduction of blurriness and over smoothing in the denoised LDCT image.

# CHAPTER SEVEN

# 7. CONCLUSION AND FUTURE WORKS

## 7.1. Conclusion

One of the best options to reduce radiation risk due to CT scans is to use LDCT, However it results in degrading elements in CT imaging by increasing the quantum noise and artifacts. As a result it lowers the visual quality of CT images, by impairing the precision of clinical assessments. A potential solution to this problem is deep Learning based LDCT restoration. Recently, with the achievement of Generative Adversarial Networks (GANs), significant progress has been achieved in LDCT restoration tasks, one of which is incorporating attention networks, a deep Learning technique for enhancing the effectiveness of LDCT restoration, to the existing deep Learning architecture, which recently gained popularity. Although the Recent attention-based deep Learning methods have gained an acceptable visual performance in CT restoration, the quantitative results of those proposed methods are not optimal in some cases, i.e. LDCT datasets with small amounts of NDCT images. This is due to a lack of adequate attention mechanisms that are given to the preservation of structural features and are provoked by appropriate cross-channel interaction. Another main issue in LDCT restoration is lack of effective pixel-wise loss functions during the model training. Recent work introduces new approach by applying dual domain discriminator to learn both global and local difference between the denoised and normal-dose images in both image and gradient domains. However, to improve the denoising quality this work must give visual attention to preserve the structural features in a better way and to encourage the generator output the denoised LDCT images that match the NDCT images, it must utilize tolerating pixel-wise loss during the model training. So, this work proposed attention based LDCT image denoising using GAN to solve the stated problems.

The goal of this study was to improve the denoising quality of LDCT by adding learning constraints to the GAN network architecture trained on the most difficult dataset, known as the Mayo-10 percent with small amounts of NDCT images - which is only 10% of NDCT of the overall dataset. To do so, this thesis work adopts an efficient channel attention block that has an adequate cross-channel communication block and adding appropriate pixel-wise loss along with other learning constraints that help to improve the performance of the neural network.

Various experiments were run side by side for comparison. In the first experiment, channel attention block and instance normalization in the generator encoder while letting the pixel-wise loss as it is in the baseline model. This experiment shows an improvement in the quality of LDCT image a little bit as shown in figure 6.3 it improves the PSNR score from 20.40010 to 21.32325 and SSIM score from 0.72109 to 0.73504. And also it improves the RMSE score from 0.09821 to 0.08902. In the second experiment, the dice loss in combination with MSE loss is introduced in to the image domain discriminator while the other constraints remain the same as in the first experiment. It improves both the PSNR score from 21.32325 to 21.76348, and RMSE score from 0.08902 to 0.08429 but, the SSIM score slightly decreased from 0.73504 to 0.73067. The third experiment uses the identical set of constraints as the first experiment but only employs the dice loss as a pixel-wise loss in the image domain discriminator. As a result, the PSNR score rises from 21.76348 to 21.81198, the SSIM score rises from 0.7306 to 0.7309, and the RMSE score drops from 0.8429 to 0.08400. This addresses the third research question of What effect would using dice loss as a pixel-wise loss have on the quality of denoised LDCT images.

In the fourth experiment, efficient channel attention block and Spatial attention layer are introduced into the generator network to generate good-quality images. This model tried to make the image sharp but there are some artifacts on the generated image sequences. SSIM score from 0.7309 to 0.73258, PSNR score from 21.81198 to 21.92848, and RMSE score from 0.08400 to 0.08286. The last experiment is the overall proposed model, and it improves the SSIM score from 0.73258 to 0.73830, the PSNR score from 21.92848 to 22.04259 and the RMSE score from 0.08286 to 0.08170. This addresses the second research question.

Finally, this study finds that using the Efficient Channel Attention Block in the generator network as an adequate cross-channel communication block and the Dice loss as a pixel-wise loss in the image domain discriminator along with instance normalization improves the quality of restored LDCT images compared to normal-dose CT images by effectively preserving structural and textural information while significantly reducing noise and artifacts.

## 7.2. Feature work

Performance is an important factor that keeps expanding in LDCT denoising. This work attempts to fill one of the many gaps in the present LDCT restoration domain by adopting cross channel interaction via an attention mechanism and tolerating pixel-wise loss in order to achieve

enhancement features in denoised LDCT images. This thesis approach does have certain restrictions, though. Due to time constraints, the proposed model could only be trained on paired NDCT and LDCT datasets that were publicly available; however, in order for it to be suitable for addressing the lack of paired data in the clinical setup, it must also be trained on unpaired LDCT datasets in an unsupervised manner. Thus, this may be viewed as an unexplored area in LDCT restoration.

The lack of structural detail preservation is a well-known problem in LDCT restoration, and this work attempts to overcome it by implementing a cross-channel communication block in the generator network. Lacking the ability to adaptively represent local properties of targeted structures in the LDCT image, however, may still lead to poor texture/detail preservation in the LDCT image. To solve this problem, the suggested solution could be further integrated with prior knowledge retrieval and representation paradigm mechanisms such as the adaptive prior features assisted restoration algorithm (APFA), which aims to improve the restoration of the LDCT images by adaptively capturing local features from NDCT scans.

For the purpose of recognizing human organs as objects, edge detection is a very helpful activity that is frequently used in the processing of medical images. Edge detection also reveals the locations of shadows and any other noticeable changes in the intensity of an image brought on by noise effects. For computational complexity, the gradients estimated by a Sobel operator in the gradient domain discriminator since it is based on the first-order derivative. So, replacing the Sobel edge detector with another edge detector such as, Canny edge detector, which is regarded as one of the best edge detectors currently in use; it ensures good noise immunity and at the same time detects true edge points with minimum error, can encourages better edge information and alleviates streak artifacts.

A further limitation of this work is that it does not assess the degree to which the proposed method can offer radiologists with confidence maps of the denoised LDCT images from the perspective of specific clinical tasks like liver lesion diagnosis. So that it might be investigated further as a potential course of action.

* * *

This research work was funded by grant number ASTU/SM-R/635/22 from Adama Science and Technology University.

# REFERENCE


Balda, M., Hornegger, J., & Heismann, B. (2012). Ray Contribution Masks for Structure Adaptive Sinogram Filtering, *31*(6), 1228–1239.

Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition, *86*(11).

Brenner, D. J., & Hall, E. J. (2009). Computed Tomography — An Increasing Source of Radiation Exposure, 2277–2284.

Cai, J., Jia, X., Gao, H., Jiang, S. B., Shen, Z., & Zhao, H. (2014). Cine Cone Beam CT Reconstruction Using Low-Rank Matrix Factorization : Algorithm and a Proof-of-Principle Study, *33*(8), 1581–1591.

Chen, H., Zhang, Y., Kalra, M. K., Lin, F., Chen, Y., Liao, P., … Wang, G. (2017). Low-Dose CT with a residual encoder-decoder convolutional neural network. *IEEE Transactions on Medical Imaging*, *36*(12), 2524–2535. https://doi.org/10.1109/TMI.2017.2715284

Chen, H., Zhang, Y., Zhang, W., Liao, P., Li, K., & Zhou, J. (2017). LOW-DOSE CT DENOISING WITH CONVOLUTIONAL NEURAL NETWORK College of Computer Science , Sichuan University , Chengdu 610065 , China National Key Laboratory of Fundamental Science on Synthetic Vision , Sichuan University , Chengdu Department of Scientific Re, 143–146.

Chen, Y., Gao, D., Nie, C., Luo, L., Chen, W., Yin, X., & Lin, Y. (2009). Computerized Medical Imaging and Graphics Bayesian statistical reconstruction for low-dose X-ray computed tomography using an adaptive-weighting nonlocal prior ☪, *33*, 495–500. https://doi.org/10.1016/j.compmedimag.2008.12.007

Conolly, S., Nishimura, D., & Macovski, A. (1986). Optimal Control Solutions to the Magnetic Resonance Selective Excitation Problem. *IEEE Transactions on Medical Imaging*, *5*(2), 106–115. https://doi.org/10.1109/TMI.1986.4307754

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. (2018). Deep learning for visual understanding: Part 2 generative adversarial networks. *IEEE SIgnal ProcESSIng MagazInE*, (January), 53–65. Retrieved from https://twin.sci-hub.tw/6638/2bc534ddaf5c9240cfff3e1bb1697682/creswell2018.pdf#page=1

Ding, Q., Long, Y., Zhang, X., & Fessler, J. A. (2018). Statistical Image Reconstruction Using Mixed Poisson-Gaussian Noise Model for X-Ray CT. Retrieved from http://arxiv.org/abs/1801.09533

Divel, S. E., & Pelc, N. J. (2019). Accurate image domain noise insertion in CT images, *0062*(c). https://doi.org/10.1109/TMI.2019.2961837

Diwakar, M., & Kumar, M. (2018). Biomedical Signal Processing and Control A review on CT image noise and its denoising. *Biomedical Signal Processing and Control*, *42*, 73–88. https://doi.org/10.1016/j.bspc.2018.01.010

Du, W., Chen, H., Liao, P., Yang, H., & Wang, G. (2019). Visual Attention Network for Low-Dose CT. *IEEE Signal Processing Lette Rs*, *26*(8), 1152–1156. https://doi.org/10.1109/LSP.2019.2922851

Du, W., Chen, H., Wu, Z., Sun, H., Liao, P., & Zhang, Y. (2017). Stacked competitive networks for noise reduction in low-dose CT. *PLoS ONE*, *12*(12), 1–15. https://doi.org/10.1371/journal.pone.0190069

Feruglio, P. F. (n.d.). Block matching 3D random noise filtering for absorption optical projection tomography, *5401*. https://doi.org/10.1088/0031-9155/55/18/009

Geraldo, R. J., Cura, L. M. V., Cruvinel, P. E., & Mascarenhas, N. D. A. (2016). Low Dose CT Filtering in the Image Domain Using MAP Algorithms. *IEEE Transactions on Radiation and Plasma Medical Sciences*, *1*(1), 56–67. https://doi.org/10.1109/tns.2016.2635131

Gerig, G., Kuoni, W., Kikinis, R., & Kübler, O. (1989). Medical Imaging and Computer Vision: An integrated approach for diagnosis and planning, 425–432. https://doi.org/10.1007/978-3-642-75102-8_64

Gholizadeh-Ansari, M., Alirezaie, J., & Babyn, P. (2019). Deep Learning for Low-Dose CT Denoising, 1–18. Retrieved from http://arxiv.org/abs/1902.10127

Goodfellow, B. I., Pouget-abadie, J., Mirza, M., Xu, B., Warde-farley, D., Ozair, S., … Bengio, Y. (2014). Generative Adversarial Networks, *27*(2), 139–144.

Goodfellow, I. (n.d.). Deep Learning.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., … Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139–144. https://doi.org/10.1145/3422622

Gu, J., & Ye, J. C. (2021). AdaIN-Based Tunable CycleGAN for Efficient Unsupervised Low-Dose CT Denoising. *IEEE Transactions on Computational Imaging*, *7*, 73–85. https://doi.org/10.1109/TCI.2021.3050266

Hashemi, S., Paul, N. S., Beheshti, S., & Cobbold, R. S. C. (2015). Adaptively Tuned Iterative Low Dose

CT Image Denoising. *Computational and Mathematical Methods in Medicine*, *2015*. https://doi.org/10.1155/2015/638568

Hitaj, B., Ateniese, G., & Perez-Cruz, F. (2017). Deep Models under the GAN: Information leakage from collaborative deep learning. *Proceedings of the ACM Conference on Computer and Communications Security*, 603–618. https://doi.org/10.1145/3133956.3134012

Hu, Z., Jiang, C., Sun, F., Zhang, Q., Ge, Y., Yang, Y., … Liang, D. (2019). Artifact correction in low-dose dental CT imaging using Wasserstein generative adversarial networks. *Medical Physics*, *46*(4), 1686–1696. https://doi.org/10.1002/mp.13415

Huang, Z., Zhang, J., Zhang, Y., & Shan, H. (2022). DU-GAN: Generative Adversarial Networks with Dual-Domain U-Net-Based Discriminators for Low-Dose CT Denoising. *IEEE Transactions on Instrumentation and Measurement*, *71*, 1–12. https://doi.org/10.1109/TIM.2021.3128703

Image, C. T., Loss, M., Loss, F., Yin, Z., Xia, K., He, Z., … Zu, B. (2021). SS symmetry.

Jadon, S. (2020). A survey of loss functions for semantic segmentation. *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, CIBCB 2020*. https://doi.org/10.1109/CIBCB48159.2020.9277638

Jia, L., Zhang, Q., Shang, Y., Wang, Y., Liu, Y., Wang, N., … Yang, G. (2018). Denoising for low-dose CT image by discriminative weighted nuclear norm minimization. *IEEE Access*, *6*, 46179–46193. https://doi.org/10.1109/ACCESS.2018.2862403

Kaiming He, Xiangyu Zhang, Shaoqing Ren, J. S. (2006). Deep Residual Learning for Image Recognition. *Indian Journal of Chemistry - Section B Organic and Medicinal Chemistry*, *45*(8), 1951–1954. https://doi.org/10.1002/chin.200650130

Kaiming He, Xiangyu Zhang, Shaoqing Ren, J. S. (2016). Deep Residual Learning for Image Recognition Kaiming. *Indian Journal of Chemistry - Section B Organic and Medicinal Chemistry*, *45*(8), 1951–1954. https://doi.org/10.1002/chin.200650130

Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2019-June*, 4396–4405. https://doi.org/10.1109/CVPR.2019.00453

Kaur, R., Juneja, M., & Mandal, A. K. (2018). A comprehensive review of denoising techniques for abdominal CT images. *Multimedia Tools and Applications*, *77*(17), 22735–22770. https://doi.org/10.1007/s11042-017-5500-5

Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–15.

Knyaz, V. A., Kniaz, V. V., & Remondino, F. (2019). Image-to-voxel model translation with conditional adversarial networks. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11129 LNCS*, 601–618. https://doi.org/10.1007/978-3-030-11009-3_37

Krizhevsky, A., & Hinton, G. E. (n.d.). ImageNet Classification with Deep Convolutional Neural Networks, 1–9.

Kulathilake, K. A. S. H., Abdullah, N. A., Sabri, A. Q. M., & Lai, K. W. (2021). *A review on Deep Learning approaches for low-dose Computed Tomography restoration*. *Complex & Intelligent Systems*. Springer International Publishing. https://doi.org/10.1007/s40747-021-00405-x

Li, M., Hsu, W., Xie, X., Cong, J., & Gao, W. (2020). SACNN: Self-Attention Convolutional Neural Network for Low-Dose CT Denoising with Self-Supervised Perceptual Loss Network. *IEEE Transactions on Medical Imaging*, *39*(7), 2289–2301. https://doi.org/10.1109/TMI.2020.2968472

Li, Z., Yu, L., Trzasko, J. D., Lake, D. S., Blezek, D. J., Fletcher, J. G., … Li, Z. (2014). Adaptive nonlocal means filtering based on local noise level for CT denoising Adaptive nonlocal means filtering based on local noise level for CT denoising, *011908*. https://doi.org/10.1118/1.4851635

Liu, Y., Ma, J., Fan, Y., & Liang, Z. (n.d.). Adaptive-weighted total variation minimization for sparse data toward low-dose x-ray computed tomography image reconstruction Adaptive-weighted total variation minimization for sparse data toward low-dose x-ray computed tomography image reconstruction. https://doi.org/10.1088/0031-9155/57/23/7923

Mao, X., Li, Q., Xie, H., Lau, R. Y. K., Wang, Z., & Smolley, S. P. (n.d.). Least Squares Generative Adversarial Networks, 1–16.

Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y., & Networks, P. (2018). S PECTRAL N ORMALIZATION.

Moen, T. R., Chen, B., Holmes, D. R., Duan, X., Yu, Z., Yu, L., … McCollough, C. H. (2021). Low-dose CT image and projection dataset. *Medical Physics*, *48*(2), 902–911. https://doi.org/10.1002/mp.14594

Mohd Sagheer, S. V., & George, S. N. (2020). A review on medical image denoising algorithms. *Biomedical Signal Processing and Control*, *61*, 102036. https://doi.org/10.1016/j.bspc.2020.102036

Öngün, C., & Temizel, A. (2019). Paired 3D model generation with conditional generative adversarial networks. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11129 LNCS*(March), 473–487. https://doi.org/10.1007/978-3-030-11009-3_29

Ouyang, L., Solberg, T., & Wang, J. (2011). Effects of the penalty on the penalized weighted least-squares image reconstruction for low-dose CBCT. *Physics in Medicine and Biology*, *56*(17), 5535–5552. https://doi.org/10.1088/0031-9155/56/17/006

Paris, S. (n.d.). A Fast Approximation of the Bilateral Filter.

Park, H. S., Baek, J., You, S. K., Choi, J. K., & Seo, J. K. (2019). Unpaired image denoising using a generative adversarial network in x-ray CT. *IEEE Access*, *7*, 110414–110425. https://doi.org/10.1109/ACCESS.2019.2934178

Sam, G. (2012). Chapter 3 – Research Methodology and Research Method. *Research Methodology and Research Method*, (March 2012), 43.

Shinde, P. P. (2018). A Review of Machine Learning and Deep Learning Applications. *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 1–6.

Shinde, P. P., Jadon, S., Brenner, D. J., Hall, E. J., Hinton, G. E., Goodfellow, I., … Vanderplas, J. (2009). Deep Learning. *International Journal of Radiation Biology*, *33*(8), 1682–1697. https://doi.org/10.1088/0031-9155/56/18/011

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–14.

Tahmasebzadeh, A., Paydar, R., Soltani-kermanshahi, M., & Reiazi, R. (2021). Lifetime attributable cancer risk related to prevalent CT scan procedures in pediatric medical imaging centers. *International Journal of Radiation Biology*, *0*(0), 1–11. https://doi.org/10.1080/09553002.2021.1931527

Tian, Z., Jia, X., Yuan, K., & Pan, T. (2011). Low-dose CT reconstruction via edge-preserving total variation regularization Low-dose CT reconstruction via edge-preserving total variation regularization. https://doi.org/10.1088/0031-9155/56/18/011

Vanderplas, J. (n.d.). [ PDF ] Python Data Science Handbook : Essential Tools For Working With Data.

Wang, J., Lu, H., Liang, Z., Eremina, D., Zhang, G., Wang, S., … Manzione, J. (2008). An experimental study on the noise properties of x-ray CT sinogram data in Radon space. *Physics in Medicine and Biology*, *53*(12), 3327–3341. https://doi.org/10.1088/0031-9155/53/12/018

Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., & Hu, Q. (2020). ECA-Net: Efficient channel attention for deep convolutional neural networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 11531–11539. https://doi.org/10.1109/CVPR42600.2020.01155

Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., Hu, Q., & Imagenet, R.-. (n.d.). Supplementary Material for " ECA-Net : Efficient Channel Attention for Deep Convolutional Neural Networks " A . Comparison of Different Methods using Here , we compare different attention methods using C . Visualization of Weights Learned by ECA Modules a, (61806140), 2–4.

Wang, Ting-Chun, Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., & Catanzaro1, B. (2018). High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. *ECS Transactions*, *25*(14), 27–35. https://doi.org/10.1149/1.3301809

Xu, Q., Yu, H., Member, S., Mou, X., & Zhang, L. (2012). Low-Dose X-ray CT Reconstruction via Dictionary Learning, *31*(9), 1682–1697.

Xu, Z., Yang, X., Li, X., & Sun, X. (2018). The Effectiveness of Instance Normalization: a Strong Baseline for Single Image Dehazing, 1–13. Retrieved from http://arxiv.org/abs/1805.03305

Yang, Q., Yan, P., Zhang, Y., Yu, H., Shi, Y., Mou, X., … Wang, G. (2018). Low-Dose CT Image Denoising Using a Generative Adversarial Network With Wasserstein Distance and Perceptual Loss. *IEEE Transactions on Medical Imaging*, *37*(6), 1348–1357. https://doi.org/10.1109/TMI.2018.2827462

Yi, X., & Babyn, P. (2018). Sharpness-Aware Low-Dose CT Denoising Using Conditional Generative Adversarial Network. *Journal of Digital Imaging*, *31*(5), 655–669. https://doi.org/10.1007/s10278-018-0056-0

You, C., Yang, Q., Shan, H., Gjesteby, L., Li, G., Ju, S., … Wang, G. (2018). Structurally-sensitive multi-scale deep neural network for low-dose CT denoising. *IEEE Access*, *6*(c), 41839–41855. https://doi.org/10.1109/ACCESS.2018.2858196

Yuan, Y., Zhang, Y., & Yu, H. (2018). Adaptive Nonlocal Means Method for Denoising Basis Material Images from Dual-Energy Computed Tomography. *Journal of Computer Assisted Tomography*,

*42*(6), 972–981. https://doi.org/10.1097/RCT.0000000000000805

Yue, Z. (2021). Rethinking the Dice Loss for Deep Learning Lesion Segmentation in, *26*(1), 93–102.

Yun, S. (2019). CutMix. *Iccv*, 6023–6032.

Zhang, J., Chao, H., Xu, X., Niu, C., Wang, G., & Yan, P. (2021). Task-Oriented Low-Dose CT Image Denoising. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12906 LNCS*, 441–450. https://doi.org/10.1007/978-3-030-87231-1_43

Zhang, Yanbo, Salehjahromi, M., & Yu, H. (2019). Tensor decomposition and non-local means based spectral CT image denoising. *Journal of X-Ray Science and Technology*, *27*(3), 397–416. https://doi.org/10.3233/XST-180413

Zhang, Yi, Wang, Y., Zhang, W., Lin, F., Pu, Y., & Zhou, J. (2016). Statistical iterative reconstruction using adaptive fractional order regularization. *Biomedical Optics Express*, *7*(3), 1015. https://doi.org/10.1364/boe.7.001015

Zhang, Yi, Xi, Y., Yang, Q., Cong, W., Zhou, J., & Member, S. (2016). Spectral CT Reconstruction with Image Sparsity and Spectral Mean, *9403*(c), 1–14. https://doi.org/10.1109/TCI.2016.2609414

# APPENDICES

## Appendix A: Ablation Study

This thesis work proposed to implement a cross-channel communication block in the generator network, the generator network is basically REDCNN and was previously used to denoise LDCT on its own. This ablation study has been made to come up to show some results on the performance of REDCNN performed solely along with MSE loss for training. Besides we also conduct the ablation study of our method to fully explore the proposed method in terms of the importance of different components, and in which combination of those components yields a high-quality image.

A total of three studies involving various combinations of each component in the proposed method were carried out. The first experiment was conducted by adding Efficient Channel Attention block in the decoder block at the final layer along with Instance Normalization in the generator encoder and with a combination of dice loss and MSE loss. And it improves the results. The second trial was performed by adding Efficient Channel Attention block before getting to the residual encoder, and adding spatial attention block between encoder and decoder along with dice loss as a per-pixel loss in the image domain discriminator. Finally, this study shows the effectiveness of the Efficient Channel Attention, Instance Normalization, and Dice loss, and adding those component as follows. In the generator encoder at each ten layers instance normalization is added, before getting to the generator encoder the input LDCT is concatenated with the output mapping of those LDCT images from efficient channel attention block and lastly, dice loss is used in the image domain discriminator gives a better quality of denoised LDCT image and maintain structural details in a better way, the over smoothness due to MSE loss in image discriminator is decreased as well.

Table A.1 Results of REDCNN

| | PSNR | SSIM | RMSE |
|---|---|---|---|
| REDCNN | 17.69981 | 0.58241 | 0.13236 |

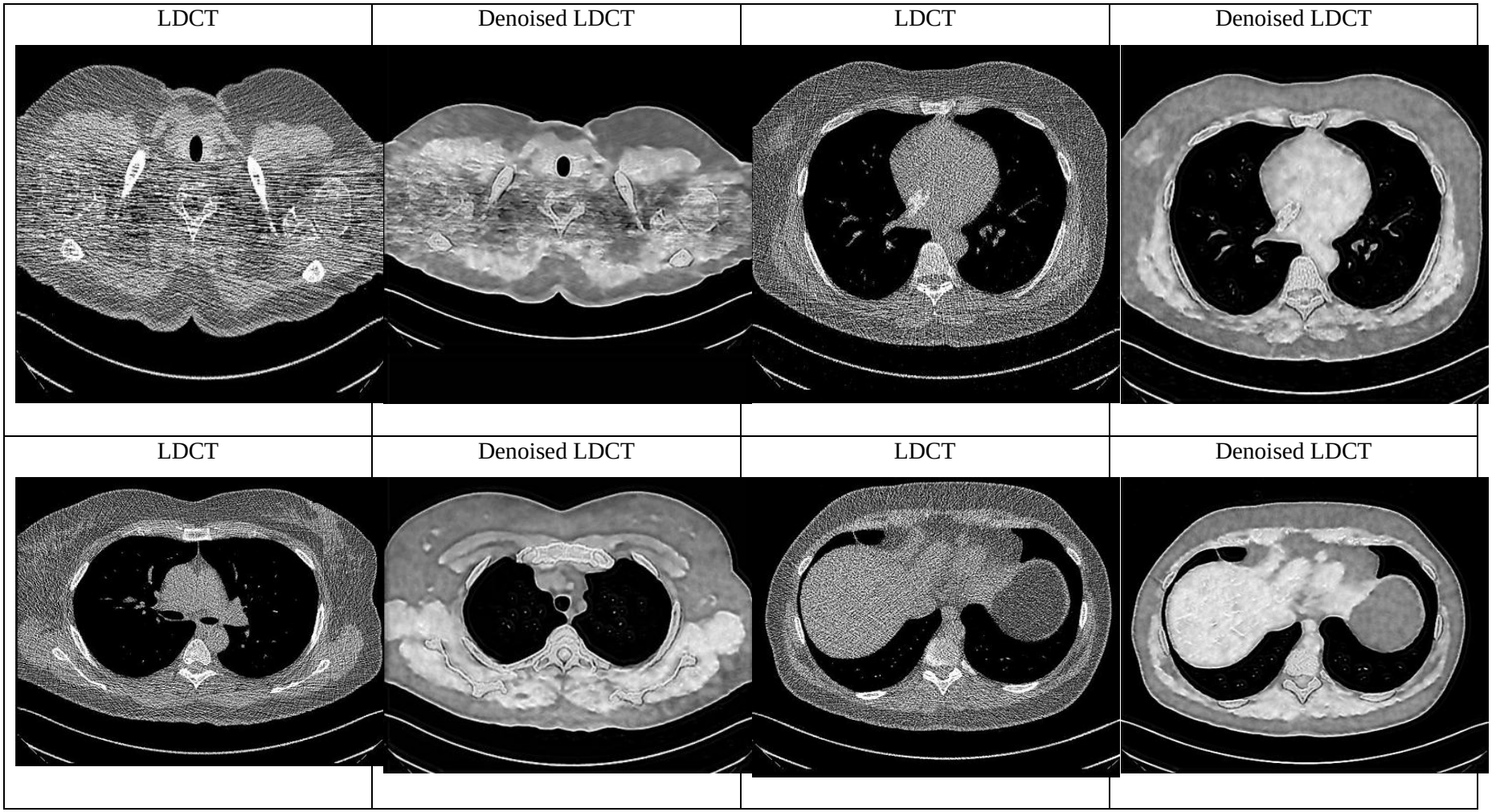
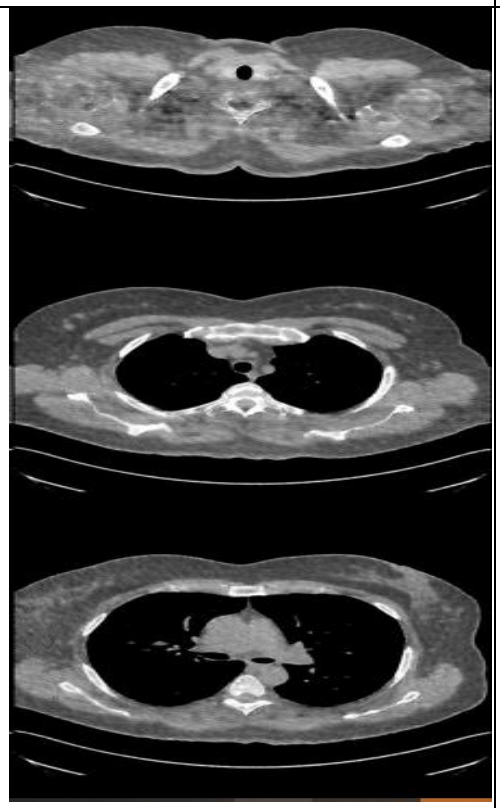
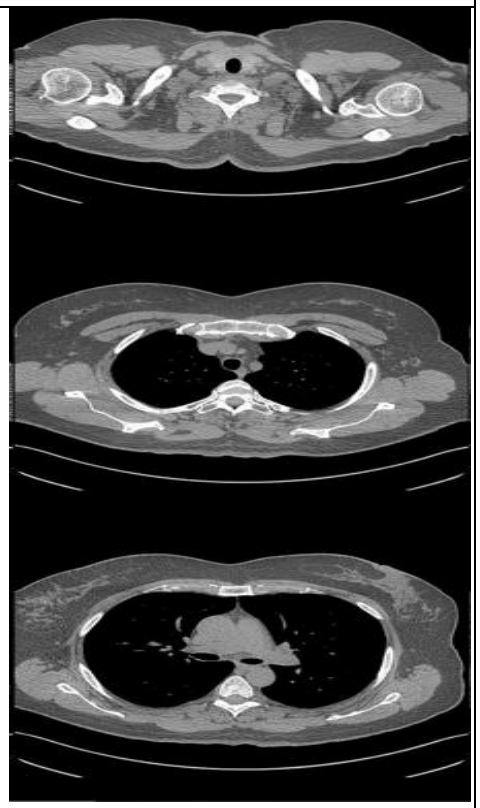
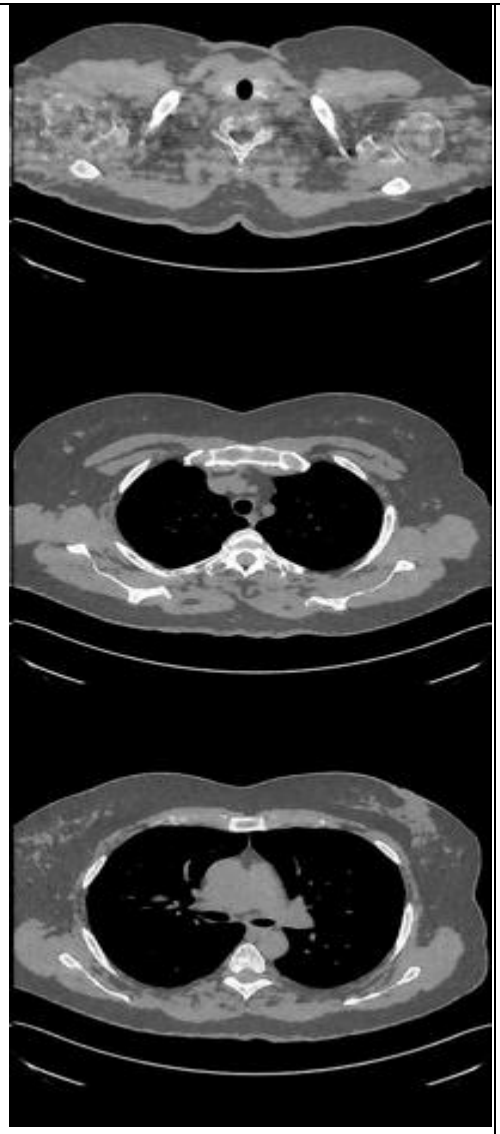


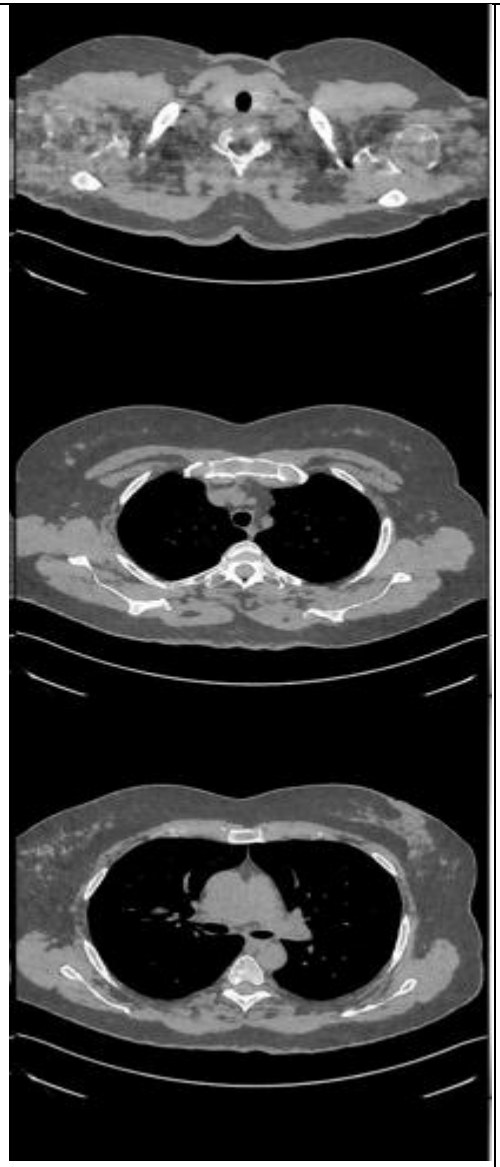
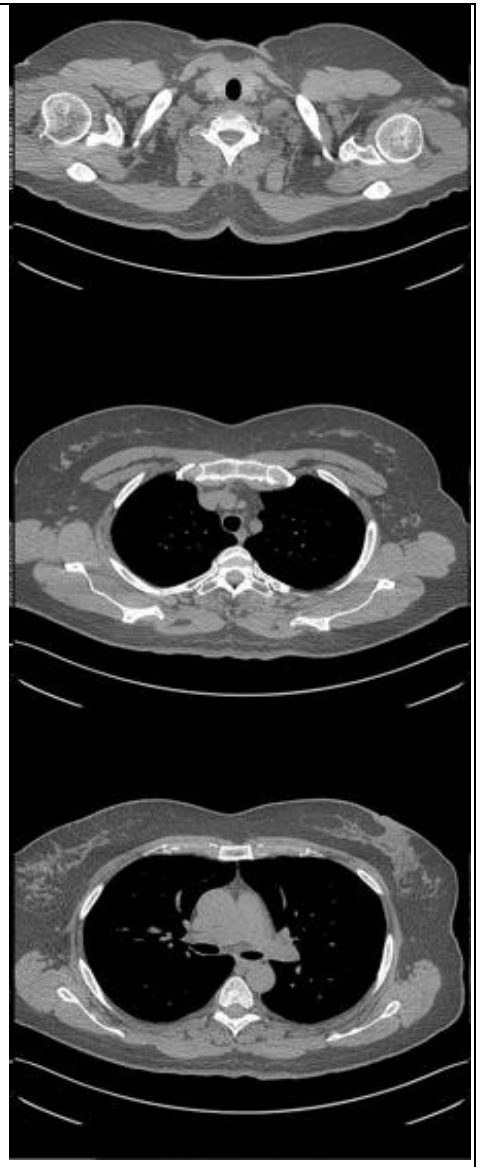

Figure A-1: input LDCT images and their corresponding denoised LDCT with REDCNN

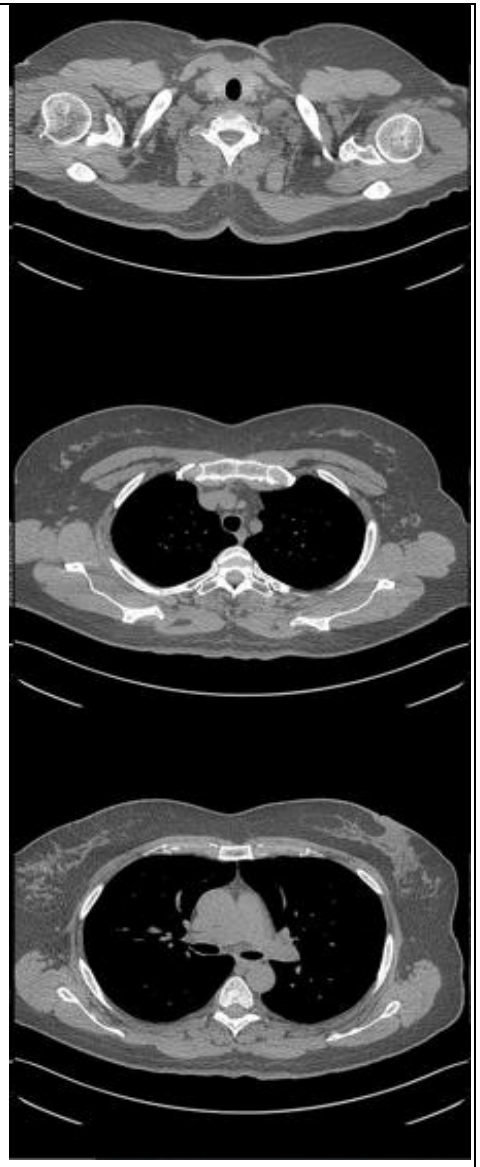
As it is clearly see in the above figure 6.13 the results from REDCNN suffers from over smoothing issue and it highly lacks structural details.

## Appendix B: Result on different iterations

| Number of iteration | Input LDCT | Denoised LDCT | NDCT |
|---|---|---|---|
| 2500 |  |  |  |

| | | | |
|---|---|---|---|
| 5000 |  |  |  |

| | | | |
|---|---|---|---|
| 7500 |  |  |  |

| | | | |
|---|---|---|---|
| 10000 |  |  |  |

## Appendix C: Sample Code

Train Generator Network

```
def train(self, inputs, n_iter):
        opt = self.opt

        self.update_moving_average()

        low_dose, full_dose = inputs
        low_dose, full_dose = low_dose.cuda(), full_dose.cuda()

        self.generator.train()
        self.img_discriminator.train()
        self.grad_discriminator.train()
        msg_dict = {}

        gen_full_dose = self.generator(low_dose)
        grad_gen_full_dose = self.sobel(gen_full_dose)
        grad_low_dose = self.sobel(low_dose)
        grad_full_dose = self.sobel(full_dose)
        self.train_discriminator(self.img_discriminator, self.img_d_optimizer,
                                 full_dose, low_dose, gen_full_dose, prefix='img', n_iter=n_iter)

        if n_iter % opt.d_iter == 0:
            ############## Train Generator ###################

            ########### GAN Loss ############
            self.g_optimizer.zero_grad()
            img_gen_enc, img_gen_dec = self.img_discriminator(gen_full_dose)
            img_gen_loss = self.gan_metric(img_gen_enc, 1.) + self.gan_metric(img_gen_dec, 1.)
            train_loss = []

            total_loss = 0.
            if opt.use_grad_discriminator:
                self.train_discriminator(self.grad_discriminator, self.grad_d_optimizer,
                                         grad_full_dose, grad_low_dose, grad_gen_full_dose, prefix=
                                         n_iter=n_iter)
```

```
            grad_gen_enc, grad_gen_dec = self.grad_discriminator(grad_gen_full_dose)
            grad_gen_loss = self.gan_metric(grad_gen_enc, 1.) + self.gan_metric(grad_gen_dec, 1.
            total_loss = grad_gen_loss * opt.grad_gen_loss_weight
            msg_dict.update({
                'enc/grad_gen_enc': grad_gen_enc,
                'dec/grad_gen_dec': grad_gen_dec,
                'loss/grad_gen_loss': grad_gen_loss,
            })
```

Train Discriminator Network

```
    def train_discriminator(self, discriminator, d_optimizer,
                            full_dose, low_dose, gen_full_dose, prefix, n_iter=0):
        opt = self.opt
        msg_dict = {}
        train_loss = []
        #y_loss = []
        ############## Train Discriminator ##################
        d_optimizer.zero_grad()
        real_enc, real_dec = discriminator(full_dose)
        fake_enc, fake_dec = discriminator(gen_full_dose.detach())
        source_enc, source_dec = discriminator(low_dose)
        msg_dict.update({
            'enc/{}_real'.format(prefix): real_enc,
            'enc/{}_fake'.format(prefix): fake_enc,
            'enc/{}_source'.format(prefix): source_enc,
            'dec/{}_real'.format(prefix): real_dec,
            'dec/{}_fake'.format(prefix): fake_dec,
            'dec/{}_source'.format(prefix): source_dec,
        })

        disc_loss = self.gan_metric(real_enc, 1.) + self.gan_metric(real_dec, 1.) + \
                    self.gan_metric(fake_enc, 0.) + self.gan_metric(fake_dec, 0.) + \
                    self.gan_metric(source_enc, 0.) + self.gan_metric(source_dec, 0.)
        total_loss = disc_loss

        apply_cutmix = self.apply_cutmix_prob[n_iter - 1] < warmup(opt.cutmix_warmup_iter, opt.cutm
        if apply_cutmix:
            mask = cutmix(real_dec.size()).to(real_dec)

            # if random.random() > 0.5:
            #     mask = 1 - mask

            cutmix_enc, cutmix_dec = discriminator(mask_src_tgt(full_dose, gen_full_dose.detach(),

            cutmix_disc_loss = self.gan_metric(cutmix_enc, 0.) + self.gan_metric(cutmix_dec, mask)
```

```
            cr_loss = F.mse_loss(cutmix_dec, mask_src_tgt(real_dec, fake_dec, mask))
            #cr_loss = focal_tversky(cutmix_dec, mask_src_tgt(real_dec, fake_dec, mask))

            total_loss += cutmix_disc_loss + cr_loss * opt.cr_loss_weight

            msg_dict.update({
                'enc/{}_cutmix'.format(prefix): cutmix_enc,
                'dec/{}_cutmix'.format(prefix): cutmix_dec,
                'loss/{}_cutmix_disc'.format(prefix): cutmix_disc_loss,
                'loss/{}_cr'.format(prefix): cr_loss,
            })

        total_loss.backward()
        train_loss.append(total_loss.item())
        #train_loss.append(total_loss.item())

        d_optimizer.step()
        self.logger.msg(msg_dict, n_iter)
        #y_loss.append(total_loss)
        #self.logger.draw_curve(total_loss, n_iter)

        #self.logger.writer_loss_discriminator(total_loss, n_iter)
        self.logger.save_losses(n_iter, train_loss)
```