#### DESIGN AND IMPLEMENTATION OF CONVOLUTIONAL NEURAL NETWORK ARCHITECTURES FOR LOW-DOSE CT IMAGE NOISE REDUCTION

by

Seyyedomid Badretale

BASc, University of Tehran, Tehran, Iran, 2010

A thesis

presented to Ryerson University in partial fulfillment of the requirements for the degree of

Master of Applied Science

in the program of

**Electrical and Computer Engineering** 

Toronto, Ontario, Canada, 2017 © Seyyedomid Badretale 2017

#### AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

## Design and implementation of Convolutional Neural Networks Architectures for Low-Dose CT Image Noise Reduction

Master of Applied Science

2017

Seyyedomid Badretale Electrical and Computer Engineering Ryerson University

#### Abstract

An essential objective in low-dose Computed Tomography (CT) imaging is how best to preserve the image quality. While the image quality lowers with reducing the X-ray dosage, improving the quality is crucial. Therefore, a novel method to denoise low-dose CT images has been presented in this thesis. Different from the traditional algorithms which utilize similar shared features of CT images in the spatial domain, the deep learning approaches are suggested for low-dose CT denoising. The proposed algorithm learns an end-to-end mapping from the low-dose CT images for denoising the low-dose CT images. The first method is based on a fully convolutional neural network. The second approach is a deep convolutional neural network architecture consisting of five major sections. The results of two frameworks are compared with the state-of-the-art methods. Several metrics for assessing image quality are applied in this thesis in order to highlight the supremacy of the performed method.

# Acknowledgements

I would like to thank my supervisor Dr. Javad Alirezaie for his great support, patience, and assistance during my master's degree and research work at Ryerson University. His trust in me let me arrive at the research topic that most suited me and make this research a valuable experience for me.

I would like to thank Dr. Paul Babyn for providing precious resources and outstanding writing support.

I am very grateful to my parents for their motivation and mental support throughout my graduate studies.

I would like to thank all my colleagues and friends for all their assistance.

# Contents

Declarationiii
Abstractiii
ist of Tablesviii
ist of Figuresix
ist of Acronymsxii
Chapter 1 Introduction 1
1.1. Background
1.2. Problem Statement
1.3. Techniques adopted
1.4. Framework
1.5. Major Contributions
1.6. Overview of this thesis4
Chapter 2 X-Ray CT Image denoising Techniques5
2.1. X-Ray Computer Tomography Imaging5
2.1.1. Fundamentals of CT imaging7
2.1.2. CT Image Quality and Radiation Dose7
2.2. CT Noise Reduction Methods
2.2.1. Sinogram Denoising
2.2.2. Total Variation Image Denoising10
2.2.3. Wavelet Thresholding for Image Denoising
2.2.4. Statistical Iterative Reconstruction Algorithms (SIR)
2.2.5. Dictionary Learning and Sparse Representation
Chapter 3 Convolutional Neural Network (CNN)

3.1. C	Convolutional Neural Network (CNN)	16
3.2.	NN components and structures	17
3.2.1.	Convolutional Layer	17
3.2.2.	Pooling layer	18
3.2.3.	Activation Function	19
3.2.4.	Loss Function	20
3.2.5.	Regularization	21
3.2.6.	Optimization techniques	21
3.3. C	Caffe	22
3.4. I	Deep learning medical applications	23
3.4.1.	Medical image classification	23
3.4.2.	Medical image denoising	24
Chapter 4	Proposed Methodologies	. 25
4.1. C	Convolutional Architecture for Low-Dose CT Noise Reduction (LDCNN)	25
4.1. C 4.1.1.	Convolutional Architecture for Low-Dose CT Noise Reduction (LDCNN)	25 25
4.1. C 4.1.1. 4.1.2.	Convolutional Architecture for Low-Dose CT Noise Reduction (LDCNN) Problem Formulation Overall Framework	25 25 26
4.1. C 4.1.1. 4.1.2. 4.1.3.	Convolutional Architecture for Low-Dose CT Noise Reduction (LDCNN) Problem Formulation Overall Framework ReLU Nonlinearity	25 25 26 28
4.1. C 4.1.1. 4.1.2. 4.1.3. 4.1.4.	Convolutional Architecture for Low-Dose CT Noise Reduction (LDCNN) Problem Formulation Overall Framework ReLU Nonlinearity Correlation to dictionary learning algorithms	25 25 26 28 29
4.1. C 4.1.1. 4.1.2. 4.1.3. 4.1.4. 4.1.5.	Convolutional Architecture for Low-Dose CT Noise Reduction (LDCNN) Problem Formulation Overall Framework ReLU Nonlinearity Correlation to dictionary learning algorithms Overall Structure	25 25 26 28 29 30
<ul> <li>4.1. 0</li> <li>4.1.1.</li> <li>4.1.2.</li> <li>4.1.3.</li> <li>4.1.4.</li> <li>4.1.5.</li> <li>4.1.6.</li> </ul>	Convolutional Architecture for Low-Dose CT Noise Reduction (LDCNN) Problem Formulation Overall Framework ReLU Nonlinearity Correlation to dictionary learning algorithms Overall Structure Implementation Details	25 25 26 28 29 30 32
<ul> <li>4.1.</li> <li>4.1.1.</li> <li>4.1.2.</li> <li>4.1.3.</li> <li>4.1.4.</li> <li>4.1.5.</li> <li>4.1.6.</li> <li>4.1.7.</li> </ul>	Convolutional Architecture for Low-Dose CT Noise Reduction (LDCNN) Problem Formulation Overall Framework ReLU Nonlinearity Correlation to dictionary learning algorithms Overall Structure Implementation Details Network parameters and Efficiency	25 25 26 28 29 30 32 34
<ul> <li>4.1. 0</li> <li>4.1.1.</li> <li>4.1.2.</li> <li>4.1.3.</li> <li>4.1.4.</li> <li>4.1.5.</li> <li>4.1.6.</li> <li>4.1.7.</li> <li>4.2. E</li> </ul>	Convolutional Architecture for Low-Dose CT Noise Reduction (LDCNN) Problem Formulation Overall Framework ReLU Nonlinearity Correlation to dictionary learning algorithms Overall Structure Implementation Details Network parameters and Efficiency eeep Convolutional approach for Low-Dose CT Image Noise Reduction (Deep-LDCNN)	25 25 26 28 30 30 32 34 37
<ul> <li>4.1. 0</li> <li>4.1.1.</li> <li>4.1.2.</li> <li>4.1.3.</li> <li>4.1.4.</li> <li>4.1.5.</li> <li>4.1.6.</li> <li>4.1.7.</li> <li>4.2. 0</li> <li>4.2.1.</li> </ul>	Convolutional Architecture for Low-Dose CT Noise Reduction (LDCNN) Problem Formulation Overall Framework ReLU Nonlinearity Correlation to dictionary learning algorithms Overall Structure Implementation Details Network parameters and Efficiency Deep Convolutional approach for Low-Dose CT Image Noise Reduction (Deep-LDCNN) Proposed network	25 25 26 28 29 30 32 34 37
<ul> <li>4.1. 0</li> <li>4.1.1.</li> <li>4.1.2.</li> <li>4.1.3.</li> <li>4.1.4.</li> <li>4.1.5.</li> <li>4.1.6.</li> <li>4.1.7.</li> <li>4.2. 0</li> <li>4.2.1.</li> <li>4.2.2.</li> </ul>	Convolutional Architecture for Low-Dose CT Noise Reduction (LDCNN) Problem Formulation Overall Framework ReLU Nonlinearity Correlation to dictionary learning algorithms Overall Structure Implementation Details Network parameters and Efficiency Proposed network Parametric Rectified Linear Unit (PReLU)	25 25 26 28 30 30 32 37 37 40
<ul> <li>4.1. 0</li> <li>4.1.1.</li> <li>4.1.2.</li> <li>4.1.3.</li> <li>4.1.4.</li> <li>4.1.5.</li> <li>4.1.6.</li> <li>4.1.7.</li> <li>4.2. 10</li> <li>4.2.1.</li> <li>4.2.2.</li> <li>4.2.3.</li> </ul>	Convolutional Architecture for Low-Dose CT Noise Reduction (LDCNN) Problem Formulation Overall Framework ReLU Nonlinearity Correlation to dictionary learning algorithms Overall Structure Implementation Details Network parameters and Efficiency Deep Convolutional approach for Low-Dose CT Image Noise Reduction (Deep-LDCNN) Proposed network Parametric Rectified Linear Unit (PReLU) Comparison between Deep-LDCNN and LDCNN	25 25 26 28 29 30 30 31 37 40 41

4.2.5.	Implementation Details
Chapter 5	Results and Discussions
5.1. C	Convolutional Architecture for Low-Dose CT Noise Reduction (LDCNN)
5.1.1.	CT Datasets
5.1.2.	Evaluation Metrics and Quality Assessment Models
5.2. D	Deep Convolutional approach for Low-Dose CT Image Noise Reduction (Deep-LDCNN) 55
5.2.1.	CT Datasets
5.2.2.	Evaluation Metrics and Quality Assessment Models57
5.2.3.	Clinical dataset
5.2.4.	Simulated dataset
5.2.5.	Architecture Robustness
5.2.6.	Computational Cost and Running Time78
Chapter 6	Conclusion and Future Works 79
6.1. C	Convolutional Architecture for Low-Dose CT Noise Reduction (LDCNN)
6.2. D	Deep Convolutional approach for Low-Dose CT Image Noise Reduction (Deep-LDCNN) 80
6.3. F	uture improvements
Reference	es

# **List of Tables**

Table 4.1. Comparison of LDCNN networks with different filter numbers    36
Table 4.2. Comparison of parameter settings and a trade-off between the parameters and performance in
Deep-LDCNN
Table 5.1. Comparison of performance of LDCNN on the thoracic CT dataset
Table 5.2.Comparison of performance of LDCNN on the CATPHAN 600 CT dataset
Table 5.3. Piglet dataset setting with four different radiation doses and 0.625 mm slice thickness
Table 5.4. The average results of quality metrics on the CATPHAN 600 in Deep-LDCNN performance.61
Table 5.5. The average results of quality metrics on the Piglet dataset in Deep- LDCNN performance 64
Table 5.6. The average results of quality metrics on the thoracic phantom dataset in Deep- LDCNN
performance

# **List of Figures**

Figure 2.1 Fundamentals of CT imaging [7]
Figure 2.2 Left: A sample CT Shepp–Logan phantom [9], Right: Corresponding sinogram 7
Figure 2.3. A wavelet approach for low-dose CT image denoising
Figure 2.4. Iteratively statistical reconstruction techniques. Coronal CT with the radiation dose of
50 mA and reconstructed with FBP method (Top left), CT reconstruction with FBP in radiation
dose of 750 mA (Top right), comparing to Advanced Statistical Iterative Reconstruction (ASIR)
in Bottom left and Model-Based Iterative Reconstruction (MBIR) in Bottom right with the
radiation dose of 50 mA in [19]13
Figure 3.1. The LeNet-5 network architecture, which was designed for the digit classification task
[22]. It consists of the Conv layers, sample pooling, Sigmoid nonlinearity and fully connected
layers and was trained on MNIST digit dataset with about 60,000 training images 15
Figure 3.2. Visualization of the Learned hierarchical features in the LeNet-5 framework [22]. Each
feature can be considered as a filter. Left is from the first layer filters of the size $6 \times 28 \times 28$ and the
right is the second conv layer filters of the size 16×10×10
Figure 3.3. Illustration of the Convolution (left) and the Deconvolution (Right)
Figure 3.4. A sample solver file for CNN implementation in Caffe
Figure 4.1. Comparison of tanh and ReLU as an activation function in LDCNN
Figure 4.2. A network structure of the end-to-end nonlinear mapping denoising LDCNN 31
Figure 4.3. Trained first layer filters in LDCNN
Figure 4.4. Comparison between five-layer and four-layer LDCNN architectures
Figure 4.5. Comparison between different filter sizes in LDCNN
Figure 4.6. The structure of the denoising Deep-LDCNN
Figure 4.7. Comparison of PreLU and ReLU on the validation set in Deep-LDCNN
Figure 4.8. Comparison between Deep-LDCNN architectures by fixing $m_1$ and $m_2$ and changing
k (Number of mapping layers)
Figure 4.9. Comparison between Deep-LDCNN architectures by fixing $k$ and changing $m_1$ and
m <sub>2</sub>

Figure 5.1. The validation convergence curve of the LDCNN network and the results of BM3D and
SSC-GSM
Figure 5.2. Zoomed images of the result of the denoising methods. Top left: BM3D, Top right:
SSC-GSM, Bottom left: LDCNN network, Bottom right: normal-dose CT 50
Figure 5.3. Visual comparison of denoising outputs on CATPHAN 600 with different contrast.
From left to right: Low-dose, Normal-dose, LDCNN, BM3D method, SSC-GSM method 51
Figure 5.4. Visual comparison of denoising outputs with on CATPHAN 600different line spacing.
From left to right: Low-dose, Normal-dose, LDCNN, BM3D method, SSC-GSM method 52
Figure 5.5. Visual comparison of denoising outputs on the Thoracic dataset. From left to right:
Low-dose, Normal-dose, LDCNN, BM3D method, SSC-GSM method
Figure 5.6. Visual comparison of denoising outputs on the Thoracic dataset. From left to right:
Low-dose, Normal-dose, LDCNN, BM3D method, SSC-GSM method
Figure 5.7. Samples of studied normal-dose CT image datasets in Deep-LDCNN
Figure 5.8. The result of the different algorithms on the CATPHAN 600 with different line spacing
in Deep-LDCNN
Figure 5.9. The result of the different algorithms on the CATPHAN 600 with different line spacing
in Deep- LDCNN performance. From left to right: Low-dose, Normal-dose, Deep- LDCNN,
BM3D method, SSC-GSM method, TV-MCA method, LDCNN, Chen CNN
Figure 5.10. The result of the different algorithms on the CATPHAN 600 with different contrast
in Deep- LDCNN performance. From left to right: Low-dose, Normal-dose, Deep- LDCNN,
BM3D method, SSC-GSM method, TV-MCA method, LDCNN, Chen CNN
Figure 5.11. The result of the different algorithms on the Piglet dataset in Deep- LDCNN
performance. From left to right: Low-dose, Normal-dose, Deep- LDCNN, BM3D method, SSC-
GSM method, TV-MCA method, LDCNN, Chen CNN
Figure 5.12. The result of the different algorithms on the Piglet dataset in Deep- LDCNN
performance. From left to right: Low-dose, Normal-dose, Deep- LDCNN, BM3D method, SSC-
GSM method, TV-MCA method, LDCNN, Chen CNN
Figure 5.13. The result of the different algorithms on the thoracic phantom dataset in Deep-
LDCNN performance. From left to right: Low-dose, Normal-dose, Deep- LDCNN, BM3D
method, SSC-GSM method, TV-MCA method, LDCNN, Chen CNN

Figure 5.14. The result of the different algorithms on the thoracic phantom dataset in Deep-LDCNN performance. From left to right: Low-dose, Normal-dose, Deep- LDCNN, BM3D method, SSC-GSM method, TV-MCA method, LDCNN, Chen CNN ...... 69 Figure 5.15. A sample image from the piglet dataset imposed on the simulated low dose and the result of the Deep- LDCNN. The first row is the normal full dose. The left column is the created low-dose with  $I_0 = 10^4$ ,  $3 \times 10^4$ ,  $5 \times 10^4$ ,  $10^5$  respectively and the right column is the output of Figure 5.16. A sample image from the thoracic dataset imposed on the simulated low dose and the result of the Deep- LDCNN. The first row is the normal full dose. The left column is the created low-dose with  $I_0 = 10^4$ ,  $3 \times 10^4$ ,  $5 \times 10^4$ ,  $10^5$  respectively and the right column is the output of Figure 5.17. A sample image from the CATPHAN 600 dataset imposed on the simulated low dose and the result of the Deep- LDCNN. The first row is the normal full dose. The left column is the created low-dose with  $I_0 = 10^4$ ,  $3 \times 10^4$ ,  $5 \times 10^4$ ,  $10^5$  respectively and the right column is the Figure 5.18. A sample image from the piglet dataset with different radiation dose and the result of the Deep- LDCNN. The first row is the normal full dose. The left column is the clinically generated low-dose with the dosage 25%, 10%, 5% respectively and the right column is the output 

# List of Acronyms

Conv: Convolution
CT: Computed Tomography
SIR: Statistical Iterative Reconstruction
TV: Total Variation
FBP: Filtered Back Propagation
ASIR: Advanced Statistical Iterative Reconstruction
MBIR: Model-Based Iterative Reconstruction
OMP: Orthogonal Matching Pursuit
BM3D: Block Matching and 3D Filtering
SSC-GSM: Simultaneous Sparse Coding and Gaussian Scale Mixture
CNN: Convolutional Neural Network
ConvNet: Convolutional Network
SGD: Stochastic Gradient Descent
ReLU: Rectified Linear Unit
LReLU: Leaky Rectified Linear Unit
PReLU: Parametric Rectified Linear Unit
RReLU: Randomized Rectified Linear Unit
DICOM: Digital Imaging and Communications in Medicine
LDCNN: Low-Dose CT Convolutional Neural Network
Deep-LDCNN: Low-Dose CT Deep Convolutional Neural Network
PSNR: Peak Signal to Noise Ratio

SSIM: Structural Similarity Index

RMSE: Root Mean Squared Error

UQI: Universal Quality Index

MSSIM: Multi-Scale SSIM

WSNR: Weighted Signal-to-Noise Ratio

VIF: Visual Information Fidelity

NQM: Noise Quality Measure

IFC: Information Fidelity Criterion

# Chapter 1 Introduction

#### 1.1.Background

X-Ray Computed Tomography (CT scan) is a method used to produce cross-sectional images from different parts of an object non-invasively and is widely used in diagnosing various medical conditions. A CT scan is a collection of X-ray images taken from various angles around a single axis of rotation. Although CT imaging is used in many application areas such as industry and archeology, the most common application of CT imaging is in medical diagnosis. The use of CT has been on the rise in many countries due to its application in a variety of diagnostic and therapeutic purposes. The quick and comprehensive insight into the patient's body noninvasively makes CT examinations significantly ubiquitous in the medical systems.

#### **1.2.Problem Statement**

A single CT scan can have 100 to 1000 times higher radiation dose than conventional x-rays. Substantial growth in the number of CT scans causes increasing in radiation dose in the patients. Moreover, the amount of x-ray needed to produce a high-quality CT image is relatively high which may result in an increasing risk of developing cancer. Moreover, a high amount of radiation dose can damage DNA, which may lead to various disease. One way of reducing the risk of radiationinduced cancer is using lower radiation dose during CT examination. However, reducing the radiation dose leads to excessive noise and image artifacts, so low dose CT images are highly noisy which makes them unreliable for diagnostic purposes. Removing noise and artifacts from CT images are considered a challenging task due to the complex patterns of noise. X-ray CT images are reconstructed using special algorithms from the X-ray projection data. The primary source of noise in X-ray projection data is the statistical fluctuations of X-ray quanta reaching the detectors, which later form a sinogram and is known the quantum noise, which can be modeled as a Poisson process. However, after the application of image reconstruction algorithms and image processing steps to the sinogram, the noise cannot be considered as a known distribution. As a result, using traditional image denoising methods are not effective in removing noise from CT images. As an example, by applying low-pass filters, high-frequency components of artifact can be eliminated but small tissues might be eliminated, and edges will be smoothed and blurred. The goal is to enhance the quality of the low-dose CT images similar to the corresponding normal-dose images. Employing deep neural networks on low-dose CT images, while preserving major content without any loss of diagnostically relevant details and underlying image structures will be an approving way to reduce the noise distortions in low-dose CT images.

#### **1.3.** Techniques adopted

Deep learning is an emerging area in AI with many successful applications in recent years. Deep learning is based on multilayer neural networks with backpropagation in which the weights of the network are adjusted iteratively based on an appropriate loss function. Deep neural networks are characterized by having a large number of layers and neurons. However, unlike traditional multi layer NNs, which are fully connected, deep neural networks have sparse connections. Convolutional neural networks (CNN) are a subset of deep neural networks. A CNN consists of a number of convolutional layers and some other optional layers such as activation and pooling. Deep learning has recently gained attention in various fields including medical image analysis such as classification, segmentation, detection, and registration. The problem of denoising can be considered as a designing task in which an appropriate filter is constructed to remove the noise. This view aligns well with CNN in which a combination of filters in different layers is learned through real examples. The nonlinear property of the overall filter lends itself to complex distributions, which otherwise are hard to design.

#### 1.4.Framework

The first proposed CNN network is comprised of four convolutional layers. The task of each layer is compared to the corresponding stage in dictionary learning algorithms. Notwithstanding the fact that using the methods that rely on learning will be more appropriate in this scenario since the noise model is learned from the actual data rather than by estimating the complicated noise model.

The second approach is proposed in order to boost the performance in terms of accelerating the network speed and improving the CT denoised image quality. The main issue regarding the adaptive and learned CT denoising technique is that the patch averaging step in the end will flat the image patches and produce the over-smoothed CT denoised image from the image patches. However, the CNN-based approaches are exhibited promising results as the last layer filter coefficients are learned through the backpropagations to modify the most optimized values. Even though, the second approach can obtain the best performance by introducing deeper structures and smaller filters. Applying additional layers capture the nonlinearity property of the image more effectively and the reduced filter banks eliminate the over-smoothing issue of the similar algorithms.

#### **1.5.** Major Contributions

Inspired by the requirements for processing the low-dose CT images and tackling the technical issues confronted and by considering an excellent performance of deep learning on image processing, our research illustrates the use of convolutional neural networks for denoising low-dose CT images.

In this thesis, two convolutional neural frameworks are proposed in order to remove noise distortions from low-dose CT images efficiently. The first approach can be considered as the

combination of the dictionary learning approaches, which is a widely used in low-dose CT image denoising, and deep learning methodologies. This correlation provides a guideline for creating the CNN architecture. For instance, a function is assigned to each of the convolutional layers in order to illustrate the role of the layer based on the corresponding stage in dictionary learning. Since the architecture is completely a feed-forward, it is not required to solve the optimization problem. It has a concise structure and adjustable design without modifying the core structure of the network.

The deep convolutional approach was proposed in order to improve the LDCNN method in twofold. First, the structure was modified to obtain the deeper framework and smaller-size filters. As the deep CNN framework tries to estimate the noise distortion, the deeper structure assists the network to capture the non-linear artifacts better. Moreover, by employing smaller-size feature maps, the over-smoothing characteristic of deep learning method was alleviated, and the results are comprised of sharp edges and more visually pleasing. Second, the number of parameters were reduced, which significantly decreased the training time and model computational complexity.

Moreover, the frameworks present end-to-end mapping solutions and all the network parameters such as the number of convolutional layers, the size of feature maps, and the number of filters in each layer were investigated to find the optimal values.

#### **1.6.Overview of this thesis**

This thesis is organized as follows. Chapter 2 provides an introduction to x-ray computed tomography imaging, the radiation dose measurement and the sources of noise in CT. A literature review on various low-dose CT image denoising is performed in this chapter as well. In chapter 3, the convolutional neural network is introduced, and the major components of the framework and libraries are illustrated in detail. Furthermore, the most prevalent deep CNN structures, which were recently published and influence the proposed low-dose CT denoting architectures are introduced. Chapter 4, provides the algorithmic methodologies thorough analysis on network parameters and experiments, which are conducted to gain the optimal settings, are illustrated. The function of each layer is studied, and the correlation between the proposed deep learning architecture and the dictionary learning approaches is demonstrated in detail. Chapter 5, contains the main results of each of the two performed convolutional networks along with the demonstration of the contributed CT. Finally, concluding summary, remarks and avenues for possible future research studies are presented in Chapter 6.

# Chapter 2 X-Ray CT Image denoising Techniques

## 2.1. X-Ray Computer Tomography Imaging

X-ray Computer Tomography (CT) is a medical diagnostic device, which is utilized to scan the inner parts of body objects based and display biological tissues by image-pixels or volumevoxels. The images are produced by changing the of points of views (angles) and based on the xray photon absorbance of different body tissues. This device has various medical diagnostic and even industrial applications. The former has been becoming a striking subject due to the effect of the x-ray photons on a human body. X-ray beams are highly-energized photons, which are capable of releasing electrons from their resting orbits and produce charged ions. These ions can be hazardous to human tissues such as DNA that can intensify the harm of cancer[1]. Several studies in CT scan imaging from 1990 to 1999, demonstrating that about 0.5% of the cancers in the United States are associated with previously CT imaging usage [2], [3].

CT scan imaging has various merits over traditional 2D x-ray radiography. 2D x-ray radiography, which was admired in the late 1990s, creates a 2D image based on the mathematical superposition of several waves of all the human tissues during scanning, in contrast to the CT imaging which produces image series, which can be inspected as image slices. As a result, the image contrast is improved and provides radiologist for inspecting the structures located inside the tissues. Researches indicated that in the North America, 120 million CT scans were conducted in 2008 [2]. Notwithstanding the fact that harm of the radiation of CT scans is negligible, by considering the popularity of performing CT scans, which each person might undergo for several times for a year, this topic has been becoming a public health concern and initiative plans need to be taken in order to decrease its risk. A wise method of decreasing the x-ray dosage absorbed by human tissues is to reduce the radiation, which accordingly generates more noise distortions and a low Signal to Noise Ratio (SNR) [4]. The concentrate of this research is to study algorithms of reducing artifacts in low-dose CT imaging.



Figure 2.1 Fundamentals of CT imaging [5]



Figure 2.2 Left: A sample CT Shepp–Logan phantom [7] , Right: Corresponding sinogram

# 2.1.1. Fundamentals of CT imaging

CT scan imaging includes a motorized bed for a patient with a revolving x-ray scanner. An x-ray generator directs high-energy photons in a fan-shaped beam to a patient and numerous detector cells in the reverse position, achieve the photons that passed through the tissues. This procedure which is indicated in Figure 2.1, lasts as the generator and the detector plane revolves, and the motorized bed is passing through the center. Subsequently, a series of image slices are created, which associated with 2D human tissue planes. Each slice series can be regarded as a 2D image-pixels or a voxel-based 3D corresponding the body tissue produced by CT software.

The raw image data generated by the detector cells is known as sinogram, which is comprised of various projection from the variety of point of views of a specific human organ. Such projections are fundamentally formed by the radon transform of the body slice. The well known Shepp-Logan [6] phantom and its radon transform are indicated in Figure 2.2. Mathematically, to generate the image, the inverse radon transform of the sinogram data have to be computed.

# 2.1.2. CT Image Quality and Radiation Dose

CT scan images are prone to various noise distortions, which are destructive to image quality. A number of elements are associated with these deformities such as patient movement during the scan, metallic implant in body organs, and inaccurate calibration of the scanner components. The most frequent noise distortions are generated by Compton scattering, photon starvation, and beam hardening. X-ray CT imaging is operated by following the fact that different body organs collect x-ray photons in a specific range of energy in various degrees. Most of the x-ray CT scan imaging use multichromatic x-ray photon beams expressing particular scope of photon energies. This matter is capable of introducing a beam hardening noise distortion, which arises when the lower energy x-ray photons are collected further than higher energy photon beams[5]. This artifact contributes as fluctuating structures in the uniform tissue structures. Notwithstanding the beam hardening noise is not large most of the time and not very conspicuous, it can get notable near metallic implants or large bone tissues[8]. Iterative and reconstructive algorithms have been presented to remove this artifact [9], [10].

Compton scattering is comprised of radiation photons, which are associating with free negative charged particles or the last orbit electrons. Consequently, the interacted photons are leaded in a divergent path and terminate in a non-aligned detector cell. This issue can introduce streak artifacts.

Generally speaking, the number of photons arrive at the detector cells is the major part, which influences the CT image quality and also the x-ray dose utilizes in conducting CT scans. Theoretically, by no means of noise distortions, the radiated photon particles arriving at the particular detector cells, follow the Poisson distribution. As a result, the insufficient number of photon counts, can introduce a low CT image quality. Because the CT scan projections transforming into pixel-based images, the artifact distortion becomes complex and undetermined and generate streak artifacts majorly along the image gradients and near the object corners.

There are some factors widely influencing the arrived photon counts. A number of them can be determined by the radiologists and the body organs of the patient. First, the voltage of the x-ray CT tube, known as a peak kilovoltage (kVp), determines the greatest energy of the discharged xray spectrum and can affect the absorbed radiations by the body organs. Second, the current of the x-ray CT tube in milliamperes (mA), which affect the x-ray signal power and the number of reached photon particles to detectors. Third, the scanning time, which defines as the exam duration and can be represented in combination with the tube current by the milliampere-seconds (mA). Fourth, the slice thickness, which is the width of the entering of the tube and influence the spatial CT image resolution. In this study, the X-ray CT properties are remained fixed, and the concentration is in the reconstructed CT image from sinogram to improve its quality. Expressing the x-ray dosage in CT can be performed by utilizing various techniques such as tissue dosage and scanner radiation output dose. Volume CT dose index  $(CTDI_{vol})$  is a prevalent method of expressing the CT output dosage. It concentrates on the radiation current emitting form the tube rather than x-ray dosage absorbing from the body tissues. Mathematically, the Energy dosage (D) which is affected by the patient age and sex, defines by the following equation:

$$D = \frac{Energy \ absorbed}{Mass \ of \ te \ corresponding \ tissue}$$
(2.1)

Where the unit of D is Gray (Gy). By employing different factors to this equation, equivalent dose and the effective dose can be proposed. In the equivalent dose, the tissue environment is taking into accounts and in effective dose, the total organs which are on the path of the emitted x-ray to the detectors, are taking into accounts.

Filtered Back Projection (FBP) is a widespread technique of acquiring a CT image from the raw scanner input. It is comprised of the imposing a sinogram by a high-pass filter and applying the inverse radon transform afterward. The first step substantially prevents the blurring noise distortions. This method, however, preserves the steak artifacts and other conventional CT noise distortions, as complicated filtering tools have been introduced in order to improve the CT image quality.

## 2.2. CT Noise Reduction Methods

Methods used for removing noise from Low-dose CT images can be divided into two main groups; denoising applied to sinogram (projection data), and denoising applied to the reconstructed image.

#### 2.2.1. Sinogram Denoising

There are a number of methodologies which attempt to affect the sinogram to create a highquality CT images. This method generally involves applying a smoothing filter on raw sinogram and imposing the FBP algorithm afterward. Another approach is applying statistical iterative reconstruction algorithms on sinogram, which iteratively compute the optimization parameters. The penalized likelihood sinogram denoising method was illustrated in [11], and poly-dimensional iterative framework was introduced in [12] in order to improve the raw projection data. Although removing noise from the sinogram is much more effective, this is not always possible since the sinogram and the raw projection data is not readily available and accessible. Therefore, in this paper, the focus is on the methods used to denoise the reconstructed image in the image space in which the noise distribution is unknown, which is referred to as CT image denoising throughout this paper.

#### 2.2.2. Total Variation Image Denoising

The total variation (TV) algorithm is a mathematical function utilized for fluctuation measurement. In image processing, it is expressed as a method to decrease the unassociated local fluctuations in an image, which was created by noise distortions. In general, the prevailing mathematical definition of TV algorithm can be written as:

$$\min_{u} \int |\nabla u(x,y)| dx dy \quad subject \ to \quad ||u(x,y) - u_0(x,y)||_2 \le \sigma$$
(2.1)

Which  $\sigma$  is the distortion variance,  $\nabla$  denotes to the gradient, u(x, y) is the desired clean image,  $u_0(x, y)$  is the noisy image. The goal of the TV in CT image denoising is to decrease fluctuations in the reconstructions process from the raw projection data by considering the minimum error. Recently, [13] proposed an improved algorithm by introducing an additional nonnegative parameter  $\lambda$  as a Lagrange multiplier in order to robust the existing TV algorithms. By increasing the variable  $\lambda$ , more detail in the reconstructed image can be seen yet preserving more oscillations.

$$\min_{u} \int |\nabla u(x,y)| dx dy + \frac{\lambda}{2} ||u(x,y) - u_0(x,y)||_2$$
(2.3)

The major benefit of this algorithm is conserving of the high-frequency components in the reconstructed image. This algorithm is also gained from the local properties and can be applied as a local image reconstruction technique by adopting Euler-Lagrange equations. Consequently, it accurately models the high-frequency components (mostly sharp edges) of the image, while preserving some low-frequency tissues from the low-dose CT images.

Since the noise in CT images are mainly localized and fluctuated, such as the white Gaussian noise and salt-and-pepper noise distortions, separation of the noise and clean image continuously, especially along the sharp edges are not feasible. As a result, by considering the oscillation nature

of the low-dose CT noise, linear isotropic diffusion TV [14] was presented to improve the image quality.

#### 2.2.3. Wavelet Thresholding for Image Denoising

The wavelet transform is widely employed in image denoising that incorporates spatialfrequency data of image structures. The discrete wavelet transform (DWT) is expressed as the discrete product of image pixels and orthogonal wavelet transform and contains the mutual correlation of CT images. It decomposes the image to a course part, which is comprised of the low-frequency information and high-frequency components of all the diagonal directions.

This approach assumes that the noisy CT image incorporates the white Gaussian noise and wavelet thresholding is applied in order to recover the sharp edges of tissues. A prevalent waveletbased algorithm is illustrated in Figure. 2.3. The low size wavelet coefficients are corresponding the noise distortion and can be reduced to a zero-centered function by applying a thresholding strategy in order to improve the CT image quality. The Bayesian function is among the suggested threshold functions. The 2D wavelet functions are utilized by generalizing one-dimensional wavelets in a way that it applied in every four directions (row, column, and diagonals) [15]. Furthermore, by introducing stationary functions and including wavelet coefficients translations with wavelet transforms, the approach became prevalent to rotational computations.

#### 2.2.4. Statistical Iterative Reconstruction Algorithms (SIR)

These approaches were developing after the filtered back propagation (FBP) method and they are widely employed by the recent CT scanner machines in order to continuously enhance the image reconstruction technique by introducing an iterative statistical framework of the noise distortions [16]. It is assumed that the arrive radiation photons are suffered from the poison noise distribution. Consequently, the emitted x-ray tube current is modelized by the Poisson noise and the linear integral is applied to capture the total noise passing through all the body organs to find the pixel value of the reconstructed CT image.



Figure 2.3. A wavelet approach for low-dose CT image denoising.

Representative statistical iterative reconstruction algorithms (SIR) applied in low-dose CT denoising are depicted in Figure. 2.4. A scanner descriptor is used to include the scanner properties, such as the geometry of the tube. These approaches involve the Taylor series expansion of the Bayesian framework for the maximum estimation of the reconstructed CT image. It introduces a significant computational cost due to extensive iterations and requires a large memory.

#### 2.2.5. Dictionary Learning and Sparse Representation

Dictionary learning algorithms have been proposed in order to alleviate the main issue of the previously performed methods, to reduce the model complexity and to discover the optimal and remarkable approach to be irrelevant to the CT scan properties. Consequently, an iterative approach was presented to modify self-activating to capture the image parameters. The dictionary learning algorithms exploited from offline computing and applied on CT images while possessing low model complexity and enhanced result on the trained dataset. It is based on applying the sparsity on region-based CT image patches by utilizing the learned dictionary atoms. The approach can be represented by the following formula.



Figure 2.4. Iteratively statistical reconstruction techniques. Coronal CT with the radiation dose of 50 mA and reconstructed with FBP method (Top left), CT reconstruction with FBP in radiation dose of 750 mA (Top right), comparing to Advanced Statistical Iterative Reconstruction (ASIR) in Bottom left and Model-Based Iterative Reconstruction (MBIR) in Bottom right with the radiation dose of 50 mA in [17]

$$\min_{D,x_i} \sum_{i} ||y_i - Dx_i||_2^2 \quad subject \ to \quad ||x_i||_0 < T$$
(2.4)

Which *D* is the learned dictionary and  $x_i$  consists of the sparse representations of the lowdose CT image region-wise pixels  $y_i$ ,  $||.||_0$  is the  $l_0$  –norm, which is the number of the non-zero entries, *T* is the threshold of the  $l_0$  –norm and this formula incorporates the non-zero atoms in the reconstructed dictionary. In the beginning, the dictionary parameters are initialized and remain constant while sparse coefficients  $x_i$  iteratively getting optimized in order to create the sparse vector. Afterwards, the sparse coefficients remain constant and dictionary atoms will iteratively update until the formula condition is satisfied.

During the sparse coding, sparse vectors from the image patches are calculated. The prevalent approach is utilizing the Orthogonal Matching Pursuit (OMP)[18]. Afterwards, by applying the formula, the dictionary components will have updated to minimize the reconstructed error and to obtain the high-quality CT image. In the proceeding stage, the denoising structure by

expressing the image as the addition of the clean image and the noise distortions, will commence. It attempts to minimize the cost function by introducing a Lagrange multiplier.

A common algorithm in low-dose CT denoising is known as block matching and filtering (BM3D) [19], which merge the block-matching with the sliding patches. The algorithm looks for the matching sliding windows within the images and is the correlated patches are stacked afterward to form 3D arrays and a suitable transform is applied to denoise the image.

The recent approach, which is known as Simultaneous Sparse Coding (SSC-GSM) [20] is utilizing the greedy sparse representations for image denoising. In order to calculate the sparse coefficients, it is iteratively modifying a group of randomly chosen coefficients and the cost functions to reduce the computational complexity.

It is worth stating that the quality of the output image using sparse coding algorithms is highly dependent on the dictionaries. As a result selecting the suitable adaptive dictionaries is a challenging task in the dictionary learning methods.

# Chapter 3 **Convolutional Neural Network** (CNN)



Figure 3.1. The LeNet-5 network architecture, which was designed for the digit classification task [21]. It consists of the Conv layers, sample pooling, Sigmoid nonlinearity and fully connected layers and was trained on MNIST digit dataset with about 60,000 training images.

## **3.1.** Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is a distinguished field in deep learning motivated by human biological vision systems. The formation of the prevailing CNN architecture can be traced to 1990s which LeNet [21] was designed for handwritten and machine-printed character recognition. It consists of the basic CNN components such as convolutional layers and back propagation learning methods. The convolutional network detected recurrent instinct patterns from image pixels. However, due to the absence of training data, the prevalent issue was the implementation of big databases. By creating a large-scale image database ImageNet, the well-known CNN framework called as AlexNet [22] was presented for ImageNet classification. It demonstrated substantial enhancements along with the deeper structure in the computer vision area. Subsequently, with arriving complex computing powers, advanced CNN structures such as VGGNet[23], GoogleNet [24], and ResNet [25]emerged in various applications. In order to get a better grasp,

The well-known convolutional natural network, the LeNet-5 framework is depicted in Figure. 3.1. This architecture is comprised of the three types of layers. It commenced with some convolutional layers. The convolution layers are designed to extract feature representations from the image through learning epochs. Each of the feature map cells is associated with the neighboring regions in the preceding layer. As a result, feature maps were produced by convolving the learned kernels with the previous layer and associated pair-wise with nonlinear activation functions. Specifically, each kernel weight is shared by all the spatial locations (sub-images) of the image. The sharing property of the convolutional structures reduce the computational complexity of the network and the number of parameters and alleviate the training process. Through presenting nonlinearity to the architecture by utilizing activation functions, CNNs become capable of capturing nonlinear structural features. Therefore, the activation functions such as sigmoid, tanh[26], and ReLU [27] apply to the convolved layer output. Consequently, the pooling layers were designed to decrease the size (resolution) of the feature maps in order to introduce the shiftinvariant kernels. The pooling layers are normally positioned among convolutional layers. Elements in the pooling layers are associated with the corresponding feature maps in the previous convolution layer. The well-known max pooling [28] was utilized in the LeNet-5 framework. By assembling several of the mentioned patterns (conv layer following activation function following

pooling layer), the designed framework is learned to capture low-level feature maps (such as curves and edges) to high-level filters (more abstract structural features) in the last convolutional layers. A sample of the learned feature maps is depicted in Figure. 3.2.



Figure 3.2. Visualization of the Learned hierarchical features in the LeNet-5 framework [21]. Each feature can be considered as a filter. Left is from the first layer filters of the size 6×28×28 and the right is the second conv layer filters of the size 16×10×10.

Following the stacked convolutional layers, fully-connected layers are usually placed in the deep neural networks especially in the computer vision tasks (detection and classification). A fully-connected layer can be thought of a single- sized convolutional layer as they associated all the elements in the preceding layer to the next layer. The output of the network is created by the last layer of the convolutional network. In the classification and segmentation task, the well-known softmax operator and Support Vector Machine (SVM) are exploited. They attempt to minimize the loss function which is predefined for the particular task through training epochs are obtaining the optimal parameters of the network. Stochastic gradient descent is the prevailing approach for the CNN optimization tasks [29].

# **3.2.** CNN components and structures

#### **3.2.1.** Convolutional Layer

In spite of the various created CNN structures, they all are comprised of the similar components. They consist of several convolutional layers. The goal of the conv layers is to learn feature patterns from the image pixels. Employing traditional neural networks in image processing area is injudicious due to an obvious logic. As an instance, for a 2D image with the size of  $300 \times 300$ , in the traditional NNs, there will be 90,000 input nodes. By considering a hidden layer of size 50,000, the size of the input parameters will be 4.5 billion. Furthermore, by increasing the

size of the network, this amount will dramatically increase. Furthermore, by applying traditional NNs, the 2D spatial structures will be ignored. Convolutional layers by definition have the advantages of local connectivity and weight sharing. Convolution layers benefit from sharing weight filters to reduce the complexity.

In some variations, they extracted feature maps become invariant to scale and rotation. As an instance, the transposed convolution which is known as deconvolution and becomes popular for visualization tasks acts as a reverse function than a traditional convolution method by associating a single input activation function with several output kernels. Specifically, in the deconvolution process, the image regions first are up-sampled by a factor of padding and strides and then associate with output by convolution. Figure. 3.3. illustrated the deconvolution technique along with the typical convolution method.



Figure 3.3. Illustration of the Convolution (left) and the Deconvolution (Right)

#### **3.2.2. Pooling layer**

Generally, convolutional layers follow by pooling layers, especially in large-scale databases. The aim is to decrease the network complexity by decreasing the amount of feature maps interconnections and filter sizes. There have been a variety of pooling layers. Max, Min, and Average pooling were the most favorable types. Max pooling which is inspired by a neuron complex cells, replacing each region components by the max value. So, it selects the highest activation output in each local region, which leads to neglected spatial invariance in the output layer. Furthermore, it decreases the size of the next layer, which leads to reducing the training parameters. Recently the generalized method called  $L_P$  pooling was introduced which relies on a feature map value and location and its functions change by the value p.  $L_P$  pooling can be formulated as:

$$y_{m,n,k} = \left[\sum_{(i,j)\in\mathcal{R}_{mn}} g_{m,n,k}{}^p\right]^{1/p}$$
(3.1)

Which  $g_{m,n,k}$  is the *k*-th feature map at position (i, j) within the pooling region  $\mathcal{R}_{mn}$ . By changing from  $p = \infty$  to p = 1, this function acts as max pooling to average pooling.

#### **3.2.3.** Activation Function

The next component of the deep neural networks is the activation functions. The main aim is to detect the nonlinear feature structure of images. The most remarkable activation function is the Rectified Linear Unit (ReLU). It is a piecewise linear function which omits the negative response of the activations. It consists of a simple max function which makes it much swifter than tanh and sigmoid functions in general. The ReLU can be represented as:

$$f_{m,n} = \operatorname{Max}(g_{m,n}, 0) \tag{3.2}$$

Where  $g_{m,n}$  is the output of the convolution or pooling layers at location (i, j).

Due to its instinct characteristic, which creates zero gradient for non-positive activations so, in backpropagations, stochastic gradient descent technique cannot update the affected parameters (weights and biases), several alternatives have been proposed in research studies afterward. The earliest option was introducing an extra constant parameter  $\lambda$  in order to alleviate the referred issue known as Leaky ReLU (LReLU). Notwithstanding its advantages,  $\lambda$  is a normalized variable which its optimal value is different in experiments and has to be determined for each examination. Formally, this function is defined as:

$$LReLU_{m,n} = Max(g_{m,n}, 0) + \lambda Min(g_{m,n}, 0)$$
(3.3)

A further technique known as Parametric ReLU (PReLU) was determining  $\lambda_i$  by stochastic gradient descent which improved the LReLU. The leaned variable  $\lambda_i$  can be updated within backpropagations along with other network parameters with trivial model complexity. This function is expressed as:

$$PReLU_{m,n} = Max(h_{m,n}, 0) + \lambda_i Min(h_{m,n}, 0)$$
(3.4)

Another presented approach was introduced, which named Randomized ReLU (RReLU). In this method, the negative area of ReLU are determined by assigning random sampled from a uniform distribution,  $\lambda_i^j$ . This approach was examined in the CNN face detection and indicated an improved result rather than ReLU. Mathematically, the RReLU is defined as:

$$RReLU_{m,n}^{J} = \operatorname{Max}(g_{m,n}^{J}, 0) + \lambda_{i}^{J}\operatorname{Min}(g_{m,n}^{J}, 0)$$
(3.5)

#### **3.2.4.** Loss Function

The principal part of the convolutional neural networks is the loss functions. The primitive approach is the Hinge Loss. It is mostly applied in classification tasks such as multi-class Support Vector Machine (SVM). Mathematically, the hinge loss can be represented as:

$$Loss_{Hinge} = \frac{1}{T} \sum_{n=1}^{T} \sum_{m=1}^{T} [Max(1 - \delta(y^{(n)}, m)g^{T}x_{n}]$$
(3.6)

Where  $y^{(n)}$  is the label indicator for each class for total T classes. Specifically, when  $y^{(n)} = m$  then  $\delta(y^{(n)}, m) = 1$ , otherwise  $\delta(y^{(n)}, m) = -1$ .  $g^T$  indicates the transpose of the network weights. *P* determine the level of the loss function. When P = 1, the function is called  $L_1$ - loss, if P = 2, the function is called  $L_2$ - loss, and so on.  $L_2$ - loss represents more flexibility than the first order hinge-loss and the studies demonstrated the supremacy of  $L_2$ - loss over the softmax and the first order hinge-loss on MNIST dataset.

A prevalent alternative loss function is a softmax loss. It can be considered as a combination of the polynomial logistic regression and the softmax. By definition, the softmax function is defined as  $s_k^n = e^{x_k^n} / \sum_{l=1}^{K} e^{x_k^n}$ , which  $x_k^n = w_k^T p^n + q_k$  is the activations of the associated layers and *w* is the network weight matrix. Considering the training set  $\{(x^n, y^n); n \in [1, N]\}$ , each input is assigned to the specific output label between 1 and *K*. As a result, softmax equation transfers the assigned output labels onto the normalized positive coefficients to form a probability distribution over the label groups. Consequently, the distributed values are utilized in calculating the softmax loss (multinomial logistic loss) based on the following equation:

$$Loss_{Softmax} = \frac{1}{N} \left[ \sum_{n=1}^{T} \sum_{k=1}^{T} 1\{y^{(n)} = k\} \log s_k^{(n)} \right]$$
(3.7)

#### **3.2.5.** Regularization

Regularization is an impactful technique in order to prevent the overfitting issue. The primitive regularization method was  $L_p$ -norm regularization. In general, regularization techniques wisely alter the loss function to avoid over fitting by introducing additional parameters to discipline the computational complexity. Mathematically, the  $L_p$ -norm regularization can be expressed by the following expression.

$$L_{p}(\theta, i, j) = \mathcal{L}(\theta, i, j) + \lambda R(\theta)$$
(3.8)

Where  $\mathcal{L}(\theta, i, j)$  is the network loss function,  $\lambda$  is the regularization parameter, and  $R(\theta)$  is the generalized  $L_p$ -norm regularization function. Specifically,  $L_p$ -norm term can be represented as  $R(\theta) = \sum_i ||\theta_i||_p^p$ . The additional parameter p controls the regularization term. When  $p \ge 1$ , the regularization term becomes convex, which alleviate the optimization technique [30]. A weigh decaying  $L_p$ -norm is a conventional term ascribed to when p = 2, which produces  $L_2$ -norm regularization. When p < 1, the  $L_p$ -norm regularization term becomes non-convex and introduces the sparse network weight filters.

### **3.2.6. Optimization techniques**

#### 3.2.6.1. Data Augmentation

Deep convolutional networks are exceedingly influenced by the sizable data. In variety of the computer vision and image processing applications, CNNs are suffering from lack of big and established training data and such as medical image processing. Data augmentation is an ingenious solution to reduce the dependability of the lack of data with the number of network parameters. This methodology is comprised of reorganizing the accessible information into an alternative dataset in the absence of modifying their constitution. Prevalent producers are consisting of uncomplicated geometric pixel-wise transformation such as mirroring, spatial shifting, angular rotating, and upsampling. The recent study introduced a novel technique in order to choose some optimum transformations among spatial geometric pixel-wise transformations [31], but it was comprised of computationally costly iterations.

#### 3.2.6.2. Network Initialization

Network initialization is substantially important in order to make the network to acquire the accelerated convergence and prevent zero-gradient issues due to the availability of large number of network parameters and non-convex characteristic of the loss function [32]. The bias parameters are usually zero-initialized, and the weight variables have to be asymmetrically initialized to improve the functionality of the hidden layers. When the deep convolutional architecture is not satisfactory initialized, either over-fitting or vanishing gradient issues will occur. A zero-mean Gaussian distribution is the most popular technique for weight initialization. Krizhevsky [22]was proposed the standard deviation of 0.001. Later on, the recent method which was called as Xavier, the mean value of the input and output of each hidden layer was suggested for the variance of the Gaussian distribution. This methodology is an adaptive method for each network convolutional layer and can prevent the saturation. The recent research [32] proposed the orthogonal weight matrix which can iteratively set the standard deviation of the Gaussian distribution.

## 3.3. Caffe

Performing convolutional neural networks from the base is a tiresome and non-essential task. For instance, employing the backpropagation method accurately necessitates computing the gradient descent of each hidden layers precisely. Subsequently, after computing the reverse propagations, it required the validation by gradient and compared to the forward results. Furthermore, effective employment of CNNs on Graphics Processing Unit (GPU) is a rigid task. As a result, employing a reliable library is a wise decision. Among all the libraries and framework available in deep learning, *Caffe* is among the established and well-structured framework, which is extensively used in a variety of research studies.

The *Caffe* framework is initiated by Berkeley AI Research (BAIR) in C++ and it uses the Compute Unified Device Architecture (CUDA) framework for implementing all the mathematical computations on GPU. It also exploits the CUDA Deep Neural Network (cuDNN) library for performing convolutional neural network such as weight matrix. Consequently, CNNS are implemented by employing cuDNN as well as CUDA on accelerated processing units. Moreover, *Caffe* has graphical interfaces and expressive architectures for MATLAB and Python.

Implementing a deep convolutional network in *Caffe* is by utilizing text files known as Protocol Buffers library. The primary text file is called as *net.prototxt* which consists of the properties of each layer such as a learning rate, a number of outputs, kernel size, strides and paddings, weight filters, biases, and activation functions and the layers have to be defined in sequence. The attached text file, which is called *solver.prototxt* as have to be provided to define the optimization algorithms. A sample solver file for CNN implementation in *Caffe* depicted in Figure. 3.4.

```
# The .net protocol buffer definition
net: "CNN_net.prototxt"
      net:
      test_iteration: 556
      # Compute testing every 500 training iterations.
test_interval: 500
# The base learning rate, momentum and the weight decay of the network.
 4
 5
     base_learningrate: 0.0001
     momentum: 0.9
weight_decay: 0
# The learning rate policy
 8
     learningrate_policy: "step"
     gamma:0.9
13
      stepsize:300000
     # Display every 100 iterations
display: 100
14
     # The maximum number of iterations
     max_iterations: 3000000
# snapshot of intermediate results
18
              5000
     snap:
    snapshot_prefix: "CNN"
# solver mode: CPU or GPU
solver_mode: GPU
20
21
22
```

Figure 3.4. A sample solver file for CNN implementation in Caffe

# **3.4.** Deep learning medical applications

#### **3.4.1.** Medical image classification

The most prominent contribution to deep learning in medical imaging is the image classification. The topic consists of classifying a medical diagnosis based on the patient medical exams. Due to the fact that medical exams are comprised of the small datasets, utilizing transfer learning methodology is a traditional approach in this area.

Transfer learning is a machine learning approach, that information attained during training in a large-scale dataset is exploited to create a pre-trained network and applied to different subjects. As a result, all the layers except a few last layers in the pre-trained network is used as a feature extractor network [33]. The prominent topic in this field is the classification of the patients whether they have Alzheimer or not, by using brain Magnetic Resonance Imaging with convolutional neural networks and softmax [34].
Another compelling topic is the lesion classification which concentrates on the specific object in the medical imaging systems. Arindra *et al.* [35] utilized local object and global structural information to implement a multi-stream network for nodule classification in chest Computer Tomography.

## **3.4.2.** Medical image denoising

Deep learning frameworks in low-dose CT image denoising, are the latest and state-of-theart techniques. Due to the complication of modeling the low-dose CT noise distortions, the existing iterative methods and dictionary learning approaches couldn't accurately remove the artifacts. Chen *et al.* [36]proposed a convolutional neural network framework consisting of three convolutional networks. The filter numbers were 64,32,1 respectively, and the size of the feature maps was 9,5,5 respectively. Another study was proposed in [37], utilized wavelet domain deep framework in order to recover the structural detail of the low-dose CT images.

Although both the presented architectures could surpass the dictionary learning and total variation techniques, they suffer from huge computational cost as the performed networks are comprised of the thousands of parameters to learn. Also, the output image was still over-smoothed, especially in the autoencoder framework due to utilizing large receptive fields.

# Chapter 4 Proposed Methodologies

# 4.1. Convolutional Architecture for Low-Dose CT Noise Reduction (LDCNN)

# 4.1.1. Problem Formulation

In low-dose CT imaging, the noise makes the edges, and fine image features blurred and distorted. The goal of low-dose CT image denoising is to map the noisy and distorted image onto the clean image such that the noise is decreased or entirely eliminated. The research has been focused on the real low-dose CT images (not interfering normal-dose CT images by Poisson or white Gaussian noise) and the corresponding normal-dose CT images.

Let *Y* be the low-dose CT images and *X* denote the corresponding normal-dose. Given a set  $\{Y^{(m)}, X^{(m)}\}_{1}^{K}$ , the aim is to learn the denoising function *F* based on the following.

The objective here is to determine the function that creates the closest image to X from noisy image Y. Function F can be considered as a nonlinear filter whose coefficients are not known but can be learned through training images. In this study, convolutional neural networks are used to learn such a filter.

Let us denote  $\{f_i\}_{i=1}^4$  as the filter size and  $\{m_i\}_{i=1}^4$  as the filter number of the four layers, the computational complexity of the presented CNN architecture can be approximately represented as follows:

$$0 \left\{ \left( f_1^2 m_1 + m_1 f_2^2 m_2 + m_2 f_3^2 m_3 + m_3 f_4^2 \right) S_{CT} \right\}$$
(4.1)

Where  $S_{CT}$  is the size of the CT image. It worth noting that the complexity is directly proportional to the size of the CT images and the formula is mostly influenced by the middle layers.

# 4.1.2. Overall Framework

The presented mapping function *F* is theoretically comprised of three stages:

- 1. **Feature Extraction:** This stage extracts sub-images (patches), in the way that patches are overlapping, from the low-dose image *Y*. Each sub-image then can be represented as high-dimensional feature maps (vector), which the length of each feature vector is the number of the feature maps.
- 2. Mapping: In this stage, each feature vector from the low-dose patch maps to another set of high-dimensional feature vector (Normal-dose features). We can consider nonlinear mapped feature vectors as the normal-dose feature vectors. Depicted vectors representing another set of feature maps.

3. **Restoration:** This stage assembles the produced normal-dose sub-images to form the output image. The patch-wise features are aggregated in order to generate a normal-dose image. We wish that this image to be equivalent to the ground truth desired normal-dose image *X*.

Next, we explain our definition of each stage.

#### **4.1.2.1.** Feature Extraction

A prevalent approach to image restoration is to extract sub-images and then represent them by feature vectors such as principal component analysis (PCA), discrete cosine transform (DCT), discrete wavelet transform (DWT), Haar wavelet, etc. We can conceptually consider this operation as convolving the image by a set of pre-trained filters (bases). In our strategy, we include the optimization of proposed filters into the optimization of the architecture. In conclusion, the first convolutional layer can be considered as follows:

$$G_1(Y) = Max(F_1 * Y + C_1, 0)$$
(4.2)

Here  $F_1$  and  $C_1$  denote the filters (bases) and biases respectively, and \* represent the convolution operation. In this context,  $F_1$  contains  $m_1$  filters of the kernel size  $n \times w_1 \times w_1$ , where  $w_1$  is the spatial filter size in the first layer and n corresponds the number of channels (RGB, YCbCr, Grayscale) of the input images. We utilize the Rectified Linear Unit (ReLU) as an activation function in the first layer after the convolution. Conceptually,  $F_1$  operates  $m_1$  convolutions on the input image so that each convolution has a size  $n \times w_1 \times w_1$ . Hence, the output is comprised of  $m_1$  feature vectors.  $C_1$  corresponds to vector of a size  $m_1$ -dimension that each component correlated to a filter.

#### 4.1.2.2. Mapping

In the output of the first layer, we have  $m_1$ -dimensional feature maps for each sub-image (patch). In the second stage, these  $m_1$ -dimensional feature vectors are mapped onto  $m_2$ -dimensional vectors. We can conceptually consider this operation as convolving with  $m_2$  filters with the spatial size  $w_2 \times w_2$ . In summary, the second convolutional layer can be considered as follows:

$$G_2(Y) = Max(F_2 * G_1(Y) + C_2, 0)$$
(4.3)

Where  $F_2$  denotes to  $m_2$  filters of the kernel size  $m_1 \times w_2 \times w_2$ , and  $C_2$  corresponds to vector of a size  $m_2$ -dimension. Theoretically, these  $m_2$ -dimensional feature maps correspond to each normal-dose patch to form an output image in the Restoration operation.

#### 4.1.2.3. Restoration

In the dictionary learning and sparse coding approaches, weighted averaging is the most prevalent method to form a final output based on the overlapping sub-images. Conceptually, the averaging acts as a pre-trained filter on feature vectors. Since the products of the proposed convolutional neural network are in the spatial domain, all the representations, which reshape to form normal-dose patches, are in the image domain and the filters can be considered as a weighted averaging. Regarding this, the ultimate normal-dose image can be considered as follows:

$$G(Y) = F_3 * G_2(Y) + C_3 \tag{4.4}$$

Where  $F_3$  denotes to *n* linear filters of the kernel size  $n_2 \times w_3 \times w_3$ , and  $C_3$  corresponds to bias of a size *n*-dimension that each component correlated to a filter.

# 4.1.3. **ReLU** Nonlinearity

Nonlinear processing as an activation function accompanied by the pooling layer makes crucial improvements to feature extraction and makes the network rotation, shift, and scale invariant. Apart from the traditional logistic sigmoid function as an activation function, a number of nonlinear rectifying functions were introduced including the absolute positive functions. Among all the activation functions, we decided to choose Rectified Linear Unit (ReLU), since several studies on deep convolutional neural networks have indicated discernible development by applying rectified nonlinear processing, and non-saturating nonlinear functions are greatly accelerated compared to saturating activation functions. By definition, the Rectified Linear Unit (ReLU) model for every input x is

$$G(x) = Max(0, x) \tag{4.5}$$

Mainly due to the non-saturating and linear form of ReLU, the convergence of the stochastic gradient descent is much faster than the hyperbolic tangent (tanh) and Sigmoid functions.

Therefore, the training time becomes noticeably shortened. Furthermore, it does not have any reciprocal and exponential computation. That is just a simple thresholding, so its implementation is much less expensive than other available alternatives. In addition, studies indicated that to acquire a high convergence rate RLU could be applied as an activation function. We have detailed our results in Figure. 4.1. Indicates that by applying ReLU a greater PSNR can be achieved. Besides, in order to get a low error (RMSE) of 0.02 in the validation set, it needs much less number of backpropagations,  $9 \times 10^7$  for ReLU as  $3.6 \times 10^8$  for tanh function on the thoracic dataset



Figure 4.1. Comparison of tanh and ReLU as an activation function in LDCNN.

# 4.1.4. Correlation to dictionary learning algorithms

In this section, we will demonstrate that dictionary learning algorithms, can be considered as deep learning structures.

As a generic rule in sparse representations approaches such as Feature-Sign or examplebased image regions are obtained from each image and mapped onto a compact dictionary. In convolutional neural network algorithm,  $w_1 \times w_1$  sub-images produced from the image by linear spatial support through the convolutions. Therefore, the first layer in thenetwork (and the normalization step as a pre-processing), can be considered as the same form of extracting patches and mapped on a low-dose dictionary.

The second stage in dictionary learning is optimization procedures of the low-dose dictionary components to produce the elements of the constitution of the normal-dose dictionary. This step can be considered as the mapping stage in ConvNets with the single size filter. Furthermore, due

to the feed-forwards and backpropagations of the gradient descent, deep learning approaches is more cost-effective than dictionary learning algorithms which rely on iteration.

In dictionary learning, the elements of the second stage mapped onto a normal-dose dictionary learning to form normal-dose sub-images. Mean operation then applied to these regions. The same concept rules in deep neural networks. Linear spatial support of the same size of the sub-images is applied to produce the normal-dose patches.

The stated confabulation proved that the deep learning is a generic term which embraces dictionary learning approaches. The only difference is exploiting learning of optimization parameters rather than iteration processes. Therefore, the proposed approach is comprised of an end-to-end mapping that all the parameters are learned effectively. It is also worth noting that as the layers are getting sparser, it is sensible to assign the number of coefficients in the second layer to be smaller than the first layer ( $m_2 < m_1$ ).

# 4.1.5. Overall Structure

The denoising convolutional neural network is a modified CNN whose its parameters, e.g. weights and biases should be learned and calculated with the backpropagation training. An overview of the framework is shown in Figure. 4.2.

The network that we have utilized in this study consists of four layers. Each of these layers consists of a number of convolution filters followed by an activation layer. The loss function is calculated at the end of every feed-forward step to update the parameters (weights and biases). The number of feature maps that are used from the first to the last layer is 64, 32, 32, and 1 respectively. The size of all convolutional filters is  $9 \times 9$ . The input of the network is the low-dose noisy image (*Y*), that passes through some convolutional layers to extract sets of feature maps to produce a denoised image F(Y). Function F denotes the mapping network which learns all the spatial weight filters and biases. The widely-used mean square error is considered as a loss function.

The loss function is calculated by stochastic gradient descent along with standard backpropagation as follows:

$$\Delta_{m+1} = \Delta_m \times 0.9 + \varepsilon \frac{\partial K}{\partial W_m^k} \cdot \ \partial W_{m+1}^k = \partial W_m^k + \Delta_{m+1}$$
(4.6)

In which *m* corresponds to each layer,  $\varepsilon$  denotes to the learning rate.





A a number of feature maps, which requires going through the activation function, form the first convolutional layer. The subsequent convolutional layers receive the output of the previous layer then pass its results through activation functions for nonlinear processing to make higher-level feature maps in order to create a corresponding noise free image. The proposed deep neural network can be implemented with the arbitrary size of images. As a result, the network trained hierarchically structured feature maps from low-level (blobs, edges, etc.) to high-level (more complex and detailed shapes). Moreover, we came up with a pre-train convolutional network, which can be used as a transfer learning.

It is worth noting that, notwithstanding the fact that in the proposed algorithm, each layer is influenced by independent and distinct intuitions. Therefore, we gather all three operations to produce a deep convolutional architecture. Besides, all the biases and filter weights have to be iteratively optimized. Regardless of the concise and simple architecture, the presented network could have achieved a great result, and it was based on the discovering of the notable correlation between dictionary learning and deep learning.

# **4.1.6.** Implementation Details

The proposed deep convolutional neural network can be implemented by using a large set of low-dose and the corresponding normal-dose CT images. Due to a lack of sample data, low-dose CT images have been created inaccurately by adding Poisson or Gaussian noise to the normal-dose CT images. We implemented our method based on real low-dose and normal-dose datasets. We adopt widely known peak signal-to-noise ratio (PSNR), root mean squared error (RMSE), and Structural similarity (SSIM) [38] as quantitative evaluation metrics which express high correlation with the human perceptual scores to compare the result F(Y) to the normal-dose image (X). Weights and biases, which are the network parameters are updated by standard stochastic gradient descent via backpropagation.

Figure. 4.3 demonstrates first layer filters which are trained on the axial thoracic CT images. We can infer that each learned filter has distinct functionality. As an example, the filters a, and d act like edge detectors at different angles. Similarly, the filters b, c, and e act like Gaussian and Laplacian filters. Throughput the network, we can observe the hierarchy structures from low-level (first order filters) to high-level feature maps.

All the input images are in Digital Imaging and Communications in Medicine (DICOM) format. The output images are also saved in DICOM format. The only pre-processing step is the normalization of all images between 0 and 1. Furthermore, we randomly cropped the training, validation and tested sets as  $33 \times 33$  pixels sub-images with the stride of 11. Therefore, we obtain several thousand created small sub-images rather than overlapping patches, and we do not input the original size images through the network.



Figure 4.3. Trained first layer filters in LDCNN

The biases are initialized to zero. The filter weights of each layer for the deep convolutional neural network are initialized by drawing randomly from a Gaussian distribution with zero mean and standard deviation 0.001. Furthermore, the learning rate for the first three layers are initially set to 0.01 and is going to decrease by a factor of 0.9 every 10000 iterations. For the last convolution layer, the learning rate is initialized to  $10^{-3}$  with the same decaying factor as the other layers, as well as the momentum term of 0.9.

The proposed algorithm is implemented using the Caffe package and Matlab R2016a. The processor is Intel core *i*7 CPU 3.4*GHz* and 16*GB* memory and the Graphics Card is GeForce GTX 1070. The training time is about 12 hours and the test time was forty seconds for each image. In the presented deep neural network, there is a compromise between speed and effectiveness. It is worth stating that even the fastest proposed model achieved better results than the other state-of-the-art algorithms. In the context of speed, the dictionary learning and the optimization algorithms are to deal with complicated optimization difficulties but the presented network has to be learned through a great number of backpropagations. The performed approach is completely flexible to modification of the parameters in order to gain better performance.

# **4.1.7.** Network parameters and Efficiency

Starting from the favored settings ( $m_1 = 64, m_2 = m_3 = 32, \forall w_i = 9, i = [1,4], )$ , some influential parameters are being changed purposefully and steadily to inspect the overall result of the network and achieve the most efficient setting.

#### 4.1.8.1. Number of layers

As comprehensively illustrated, deep convolutional neural networks can produce better results by increasing the number of layers. Therefore, another framework is being examined by adding an extra layer to the existing architecture. By determining the parameters of an additional non-linear layer, number of filters could be  $m_5 = 32$  or  $m_5 = 16$ . The first option add a significant complexity to the network due to the equation 4.1. Hence, it would be judicious to stick with the latter choice. Another parameter to determine is the filter number which could be  $w_5 = 9$  or  $w_5 = 7$ . To come to the point, we examined 9-9-9-9, 9-9-9-7-9,9-7-7-9-9,9-7-7-9. Other parameters such as learning rate are remained the same. In the context of speed (training time), it is evident in Figure. 4.4. that adding an additional layer will reduce the convergence speed of the architecture. By considering the performance, both settings will roughly converge to the same number in which the 4-layers network is comprised of much less parameters. We also noticed that bigger architecture not necessarily results in better performance unlike the image classification (ImageNet challenge) as in Figure 4.5. the result of the 5-layers network decayed and couldn't exceed the corresponding 4-layers architecture. This may cause by the structure of the network which doesn't consist of full connected layers and large-scale datasets.



Figure 4.4. Comparison between five-layer and four-layer LDCNN architectures

#### 4.1.8.2. Number of feature maps

On the whole, a network result should ameliorate by expanding its size such as appending more feature vectors. By considering the initial filter numbers in the proposed architecture was  $m_1 = 64, m_2 = m_3 = 32$ , we implemented two settings. First was expanding the network to  $m_1 = 128, m_2 = m_3 = 64$ . Second was shrinking the architecture to  $m_1 = 32, m_2 = m_3 = 16$ . The network was trained on the real thoracic phantom CT dataset. The performance is taken at  $8 \times 10^7$  backpropagations and illustrated in Table 4.1. As we expected the greater result was attained by expanding the filter numbers to the expense of speed (training time). As the initial network contained around 256,000 parameters and the expanded network was consisted of almost 1,000,000 parameter sand shrined network was comprised of nearly 66,000 parameters to train. Therefore, the computational complexity of the network would increase by four times by implementing the larger network but it achieved the superior result.

	$m_1 = 12$	.8,	$m_1 = 64,$			$m_1 = 32,$		
$m_2 = m_3 = 64$			$m_2 = m_3 = 32$			$m_2 = m_3 = 16$		
PSNR (dB)	RMSE	Parameters	PSNR (dB)	RMSE	Parameters	PSNR (dB)	RMSE	Parameters
32.14	0.16	1,000,000	32.11	0.19	256,000	32.01	0.27	66,000

Table 4.1. Comparison of LDCNN networks with different filter numbers

#### 4.1.8.3. Filter size

The performance of the network is explored by modifying the feature vector sizes. By and large, by increasing the feature vector sizes, deeper referred neighborhood information is collected, that successively superior performance is achieved. The default settings of the network was set to the filter sizes  $\forall w_i = 9, i \in [1:4]$ , abbreviated to 9-9-9-9. First, the mapping layer filter sizes were modified from 9 to 11 and 7 (9-11-11-9 and 9-7-7-9). The performance is remarkably improved by increasing from 7 to 9 and marginally enhanced from 9 to 11. The investigation was extended to modifying the filter size of the feature extraction layer from 9 to 7 and 11 (11-9-9-9 and 7-9-9-9.) The result was faintly improved by increasing the first layer filter sizes. Finally, the filter size of the restoration layer was altered and the same pattern was achieved as the previous experiment.

As a result, the most significant impact of filter numbers was in the mapping layer which increased filter numbers were encompassed more extensive neighborhood information that generated superior performance. It is also worth noting that increasing the filter number would enhance the computational complexity which in turn increases the training time. Therefore, selecting the wider network which the performance was improved insignificantly but it involved much more parameters over the optimal network is not cogent.



Figure 4.5. Comparison between different filter sizes in LDCNN

# 4.2. Deep Convolutional approach for Low-Dose

# **CT Image Noise Reduction (Deep-LDCNN)**

# 4.2.1. Proposed network

The proposed algorithm is designed to learn an end-to-end mapping function F between lowdose images and medium-dose images. The overall structure consists of eight convolutional layers. The recommended deep learning architecture is inspired by the sparse coding algorithm for denoising low-dose CT images and the deep learning structures in image processing. As illustrated in Figure. 4.6. the network comprised of five parts- feature extraction, compressing, mapping, enlarging, and reassembling.

# 4.2.1.1. Feature extraction

With the same approach as in LDCNN, the first convolutional layer can be considered as the convolution of a set of filters with the input image (Y) then adding with the biases. This stage extracts sub-images (patches), in the way that patches are overlapping, from the low-dose image Y.

# 4.2.1.2. Compressing

Due to the fact that the computation complexity of the network relies on the filter numbers of the two following layers and the filter size to the power two, placing another layer with lower filter size between mapping and feature extraction layers is advantageous in order to reduce the computation complexity of the network.

# 4.2.1.3. Mapping

Heretofore, we have multi-dimensional feature maps for each sub-image (patch). In the third stage, these multi-dimensional feature vectors are mapped onto other multi-dimensional vectors. This convolutional layer can be considered as the convolution of a number of filters with the output of the second layer then adding with a set of biases. Theoretically, these multi-dimensional feature maps correspond to each normal-dose patch to form an output image in the reassembling operation.

## 4.2.1.4. Enlarging

As the number of feature vectors is reduced by the compressing layer, enlarging layer is adopted in order to expand the number of filters to boost the performance of the network. This step can be considered as a reverse procedure of the compressing step. This layer can be considered as a well-known pooling layer, but it was employed by convolutional kernels. As a result, the weights are updated in each iteration and can produce a better result than a traditional pooling layer.

## 4.2.1.5. Reassembling

With the same approach as in LDCNN method, the ultimate normal-dose image can be considered as the convolution of a number of filters with the output of the second layer then adding with a set of biases.

This stage assembles the produced normal-dose sub-images to form the output image. The output is desired to be an equivalent to the ground truth desired normal-dose image X.

It is worth noting that, each network layer in the Deep-LDCNN is influenced by independent and distinct intuitions. Therefore, we gather all five operations to create a deep convolutional architecture. Besides, all the biases and filter weights have to be iteratively optimized. Each layer follows an activation function which is Parametric Rectified Linear Unit (PReLU) in the proposed research. All the parameters are updated after calculating the loss function in each iteration and during the backpropagation. The feature vectors obtained in each convolution layer from the first to the last layer are 64,16,16,16,16,16,64, and 1 respectively. The size of convolutional filters is 9,1,3,3,3,3,1,5 respectively. It is also worth stating that, the proposed convolutional neural network created low-level feature vectors such as edges following hierarchically high-level feature maps (more complicated and detailed shapes).

Let us denote  $\{f_i\}_{i=1}^7$  as the filter size and  $\{m_i\}_{i=1}^7$  as the feature number of the network layers, the computational complexity of the presented CNN architecture can be approximately represented as follows:

 $O\left\{\left(f_{1}^{2}m_{1}+\sum_{i=1}^{5}m_{i}f_{(i+1)}^{2}m_{(i+1)}+m_{6}f_{7}^{2}\right)S_{CT}\right\}$ 

(4.7)

Figure 4.6. The structure of the denoising Deep-LDCNN.

Where  $S_{CT}$  is the size of the CT image. The parameters of PReLU was neglected due to its insignificant computational cost.

It is worth noting that the complexity is directly proportional to the size of the CT images and the formula is mostly influenced by the middle layers.

# 4.2.2. Parametric Rectified Linear Unit (PReLU)

PReLU was invented to alleviate the issue of the well-known Rectified Linear Unit (ReLU). By definition, ReLU is a piecewise linear function that preserves the positive side and cut off the negative areas. Due to its non-elaborated operation, a computational process would be much faster than tanh or sigmoid activation functions. However, due to its pruning structure, ReLU has a probable drawback which is every time the function is inactive, the gradient becomes. This point can effect on some units so that they will not contribute to the optimization to adjust the filter weights.

Regarding this non-trivial issue, Parametric ReLU has been introduced by He *et al.*[39]. This activation function exploits the advantage of an extra parameter  $\lambda$  and it defines as:

$$F(x_{i,j}) = Max(0, x_{i,j}) + \lambda Min(0, x_{i,j})$$

$$(4.8)$$

Which  $\lambda$  is adopted as a learned parameter for the negative part which can be trained together with other parameters from training data by adding a trivial computational cost. Compared with ReLU, the Parametric ReLU has an advantage of producing small, non-zero gradients by possessing a non-zero negative part. The result of applying PReLU and ReLU on the thoracic dataset is shown in Figure. 4.7.



Figure 4.7. Comparison of PreLU and ReLU on the validation set in Deep-LDCNN.

# 4.2.3. Comparison between Deep-LDCNN and LDCNN

The first algorithm is named Low-dose Convolutional Neural Network (LDCNN) and the second algorithm is called Deep-Low dose Convolutional Neural Network (Deep-LDCNN). The presented network has better performance to the previous network in two ways. First, we improve the performance of the network by obtaining a deeper structure in producing higher SSIM and PSNR outputs. Second, we accelerate the performance by decreasing the number of parameters to learn. It proved that the performance of the network can be improved by replacing a single wide layer with several narrower layers with fewer feature maps. The following details are explaining these modifications. In the first layer, as the number of filters in the first layer (low-level feature maps) is sizable, it makes the computational complexity of the network significantly high. Therefore, in order to boost the speed of the methodology, Lin et al. [27] suggested adding another layer with single-size feature maps. By following the same concept, the compressing layer is connected to the first layer of the network (patch extraction) with single-size feature maps. Furthermore, as achieving sparser image descriptors by moving to next layers, the number of outputs in this layer should be less than the first layer. The next two layers in LDCNN which can be considered as mapping layers with feature size 9 (from Low-dose to normal-dose), modified into four layers with feature size 3. As mentioned earlier, this part is the most influential to the network complexity. In spite of the fact that we tried deeper networks in LDCNN, there were not enough experiments on producing more efficient outcomes due to not modifying the depth and the width of the mapping layer simultaneously. During experiments on Deep-LDCNN, we could reach to the balance between the number of layers (depth) and the number of feature maps (width). We inputted the width to be three as a medium filter size for all the layers in this part for consistency. Next step was adding an enlarging layer with an inverse approach of the compressing layer. Due to limiting the feature maps, the output would be a low-quality. Thus, enlarging feature maps after the mapping layer would considerably improve the image-quality (by 0.5dB in [22]). With the consistency with the compressing layer, filter sizes would be  $1 \times 1$ . The issue last but not least, the last layer (reassembling layer) variables were altered by a fewer outputs and feature maps. As a result, we decreased the number of parameters from 256000 in the current network to 18000 in the proposed network which is an outstanding improvement in the training time. We also utilize the parametric Rectified Linear Unit (PReLU) as an activation function instead of widely-known ReLU due to null features caused by zero gradients in ReLU. In summary, the new architecture is entirely symmetric, and it is narrow in the middle and wide at the ends.

In order to the idea of elaborating the earlier convolutional network to achieve the Deep-LDCNN, the process of the transformation was presented in three steps. First, in the reassembling layer, filter sizes were changed from 9 to 5. This step accelerates the network by 1.5 based on the equation 4.1. Second, the two nonlinear mapping layers in LDCNN are transformed to the combination of the four mapping layers with fewer parameters, a compressing layer beforehand and an expanding layer afterward. Altogether, the parameters were reduced from 190000 to 18000. Moreover, the architecture obtained a speed up of 8.2. This achievement proves that the depth of the network is the prominent factor in boosting the performance. Therefore, hick mapping layers in LDCNN were replaced with six thin layers in Deep-LDCNN. Hitherto, the performance was strengthened by 1.7 dB. As a result, the performance (average PSNR) was enhanced by 1.9 dB, and the speed of the network has been enhanced by 18.3 by providing evident and comprehensible steps. It is also worth stating that the Deep-LDCNN outperformed LDCNN both in speed and performance by a large margin. The performance was achieved by training on the CATPHAN600 dataset.

# 4.2.4. Network parameters and Efficiency

In order to validate the proposed Deep-LDCNN structure and to find the optimal settings for the best performance, a number of experiments have been managed. Due to the general rule in deep convolutional neural networks, it is unfeasible to analyze all the network parameters; mainly feature map size and number of filters in each layer. As a result, the most dominant parameters which have the most significant impact on the network efficiency have been identified and other parameters would leave unaltered. These sensitive parameters have been investigated which their minor adjusts would notably influence the performance of the network. By preserving the symmetric property of the network since it is narrow in the middle and wide at the ends, like an hourglass, all the layers which were considered as the mapping layers, kept identical. Moreover, the compressing layer and the enlarging layer had the reverse structure. Therefore, the number of filters and the number of channels in the compressing layer corresponded the number of channels and the number of filters in the expanding layer respectively. As a result, three variables have been investigated as sensitive parameters—the number of feature maps in the patch extraction layer,  $m_1$ , the number of filters in the compressing layer,  $m_2$ , and the number of mapping layers, k. As a consequence, the overall framework of the network which consists of five parts can be noted by the following formula:

$$Conv (9, m_1, 1) - PReLU (9, m_1, 1) - Conv (1, m_2, m_1) - PReLU (1, m_2, m_1) - k \times \\ [Conv (3, m_2, m_2) - PReLU (3, m_2, m_2)] - Conv (1, m_1, m_2) - PReLU (1, m_1, m_2) - \\ Conv (5, 1, m_1)$$

$$(4.9)$$

Furthermore, the computational complexity of the proposed network computed as:

$$O\left\{\left(w_{1}^{2}m_{1}+\sum_{i=1}^{5}m_{i}w_{(i+1)}^{2}m_{(i+1)}+m_{6}w_{7}^{2}\right)S_{CT}\right\} \to O\left\{\left(9^{2}m_{1}+m_{2}m_{1}+k\left(3^{2}m_{2}^{2}\right)+m_{1}m_{2}+5^{2}m_{1}\right)S_{CT}\right\} = O\left\{\left(2m_{1}m_{2}+9km_{2}^{2}+106m_{1}\right)S_{CT}\right\}$$

$$(4.10)$$

In order to validate the proposed Deep-LDCNN structure and to find the optimal parameters for the best performance, a number of experiments regarding the selected sensitive parameters have been managed. Table 4.2. Summarize all the twelve experiments which have been conducted. The sensitive variables were set to  $m_1 = 56,64$ ,  $m_2 = 12,16$ , and k = 3,4,5. Average PSNR was used as a quantitative criterion and the anthropomorphic thoracic phantom (TCIA) was used as a CT dataset. First, by considering the fact that deep convolutional neural networks can produce better results by increasing the number of layers, the effect of the number of layers was been studied. Therefore, the value of k was modified while  $m_1$  and  $m_2$  were remain fixed. The result was not unexpected and was aligned with the CNN general rule. k = 5 produced higher PSNR rather than k = 4 and k = 3. The performance was indicated in Figure. 4.8. Notwithstanding, we should take it into account that adding and extra layer will introduce more complexity to the network and will reduce the convergence rate. Furthermore, the trend between k = 3 and k = 4was more considerable than between k = 4 and k = 5.

In another parameter settings, the performance of the network is explored by modifying the sensitive variables  $m_1$  and  $m_2$ . On the whole, a network result should ameliorate by expanding its size such as appending more feature vectors and filter sizes. On the contrary, it will include more parameters in training stage which will increase the training time. The results were indicated

in Table 4.2. and Figure. 4.9. compared the results of the high-ranked parameter settings. As it is obvious, the improvements in k = 5 was marginal rather than k = 4. By and large, there is a balance between the number of parameters and the network performance. It is also with stating than in all the parameter settings, the results were better than LDCNN and, the parameters were reduced at least.



Figure 4.8. Comparison between Deep-LDCNN architectures by fixing  $m_1$  and  $m_2$  and changing k (Number of mapping layers)

Table 4.2.	Comparia	son of	parameter	settings and	l a trade	e-off between	n the pa	arameters	and pe	rforman	ice
				in Deep-l	LDCNN						

Parameter Settings	$m_1 = 56$		$m_1 = 64$		
	$m_2 = 12$	$m_2 = 16$	$m_2 = 12$	$m_2 = 16$	
k = 3	34.134	34.167	34.178	34.187 (15,744 parameters)	
k = 4	34.223	34.238	34.239	34.3 (18,048 parameters)	
<i>k</i> = 5	34.302	34.305	34.306	34.306 (20,352 parameters)	



Figure 4.9. Comparison between Deep-LDCNN architectures by fixing k and changing  $m_1$  and  $m_2$ 

# 4.2.5. Implementation Details

By training, the network parameters such as weights and biases in the regression objective were optimized via backpropagation by standard stochastic gradient descent. The presented deep convolutional neural network required sets of low-dose CT images as an input and the corresponding normal-dose as an output. Ascribable to insufficient sample data in medical imaging, low-dose CT images were generated imprecisely by imposing White Gaussian noise or Poisson on the normal-dose CT images which make the approach unrealistic. However, the performed research relied on real low-dose and normal-dose datasets.

It is worth noting that all the images were cropped to achieve several thousand sub-images to train the deep network. The biases were set to zero initialed. The filter weights were initialized by producing vectors randomly from a Gaussian distribution with zero mean and standard deviation 0.001. Moreover, the learning rate of all the layers except the last layer was initiated to 0.01 was lessening in every 10000 iterations by a factor of 0.9. For the last layer, the learning rate is initialized to 0.001 with the same decaying factor as the other convolutional layers, and the momentum was set to 0.9.

# Chapter 5 Results and Discussions

# 5.1. Convolutional Architecture for Low-Dose CT Noise Reduction (LDCNN)

The proposed algorithm was validated by experiments using phantom CT datasets, and the results are compared with BM3D [19] and Simultaneous Sparse Coding (SSC-GSM) [20]which is claimed as the state-of-the-art image restoration by the authors.

We randomly shuffle images in each dataset in order to reflect enough diversity in the training, validation, and test data. The training, validation, and test data include 50%,20%, and 30% of each dataset respectively.

# 5.1.1. CT Datasets

In order to certify the performance of the proposed architecture, a widely used CT phantom named CATPHAN600 was utilized in this experiment. This prevalent standard phantom consists of 583 images which images involve line pairs of different spacing and spheres with varying contrast and is used for evaluating spatial resolutions. All the images have the size of 512×512 and the slice thickness of 0.625 mm. Normal-dose images have the exposure rate of 210 mA, and the one with the 60 mA can be considered as low-dose CT images. The convolutional neural network was trained with the portion of the normal-dose and low-dose image pairs and the rest of the CT images considered in validation or testing phase.

In Another dataset, 407 normal-dose and the corresponding low-dose CT images were taken from The Cancer Imaging Archive (TCIA)[40]. All the images have the size of 512×512 and the slice thickness of 0.75 mm. The exposure rate of normal-dose is 200 mA, and the low-dose is 25 mA. This dataset was focused on the large nodules and put efforts on factors which influence the precision and accuracy of nodule size detection, and the images were taken with the Siemens 64row scanner. In this anthropomorphic thoracic phantom, artificial nodules with different characteristics (size, density, shape, location) were inserted and studied.

# 5.1.2. Evaluation Metrics and Quality Assessment Models

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

The long-established and most favorable image evaluation technique is the peak signal-tonoise ratio (PSNR). In this algorithm, the noisy image (target) is compared pair-wise to the reference image by pixel. Mathematically, the mean square error is initially computed by averaging the squared intensity pixel-wise differences ( $L_2$ -norm) of the distorted image and the reference image. As a result, MSE can be represented as:

$$MSE = \frac{1}{PQ} \sum_{i=1}^{P} \sum_{j=1}^{Q} (m_{ij} - n_{ij})^2$$
(5.1)

Which  $m_{ij}$  and  $n_{ij}$  are the pixel values at the location (i, j) in the noisy and source images and P, Q is the size of the image. Afterwards, the PSNR is calculated by considering the maximum value of the pixels (which is one in this study since we normalized all the CT images). PSNR can be calculated by the following formula:

$$PSNR = 10 \times \log\left(\frac{1^2}{MSE}\right) dB \tag{5.2}$$

Where the images were normalized so the gray level intensities are in the range [0,1] and the maximum gray level is 1.

The popularity of this method is due to its high accessibility, possessing a comprehensible physical definition, being independent of the viewing conditions, and the capability to evaluate the white noise. Notwithstanding, a number of recent studies declared the ineffectiveness this method on the wide range of noise distributions. Furthermore, this algorithm is not aligned with the human visual perception due to its dependencies to the energy of errors instead of structural distortions.

## 5.1.2.2. Structural Similarity Index (SSIM)

An improved method for evaluating images is the similarity index (SSIM) [38]. Images are significantly structured. The image pixels indicate high correlation, especially in the neighbor regions and these dependencies exhibit precious information about the objects in the visual scene. It indicates that the image noise distortions can be discerned by determining the degraded structural patterns information. SSIM was introduced based on comparing the local pattern structures of normalized pixel intensities by contrast and luminance. SSIM can be computed as the following:

$$SSIM = \frac{(2\mu_i\mu_j + C_1)(2\sigma_{ij} + C_2)}{(\mu_i^2 + \mu_j^2 + C_1)(\sigma_i^2 + \sigma_j^2 + C_2)}$$
(5.3)

Which  $C_1 = (K_1L)^2$ ,  $C_2 = (K_2L)^2$ , *L* correspond to the grayscale pixel intensities of the source images and  $K_1$  and  $K_2$  are the positive constant parameters.  $\mu_i$  and  $\mu_j$  are the mean values, and  $\sigma_i$  and  $\sigma_j$  are the standard deviations of the distorted and source images respectively. SSIM has been applied to various noise distortions such as mean shift, contrast stretching, speckle noise and Gaussian white noise and exhibit promising results.

Table 5.1. indicates the average denoising results for the proposed deep convolutional neural network algorithm, BM3D, and SSC-GSM for the thoracic CT dataset. We include the same results for the CATPHAN 600 phantom data in Table 5.2. As it is evident in Tables 5.1, and 5.2, the proposed deep learning method provides the highest scores in all quantitative metrics. It is worth noting that the result of our proposed method is based on the output from  $10^8$  backpropagation. Thus, it has the best performance amongst all the mentioned methods.

Table 5.1. Comparison of performance of LDCNN on the thoracic CT dataset

Eval. Mat Proposed Algorithm		SSC-GSM	BM3D
PSNR	32.112	28.754	28.298
SSIM	0.834	0.786	0.736
RMSE	0.248	0.365	0.385

Table 5.2. Comparison of performance of LDCNN on the CATPHAN 600 CT dataset

Eval. Mat	Proposed Algorithm	SSC-GSM	BM3D
PSNR	41.234	38.967	38.578
SSIM	0.941	0.913	0.919
RMSE	0.008	0.011	0.011

Figure. 5.1. Demonstrates how the number of backpropagations can improve the result of our proposed LDCNN method. It is worth noting that after the  $7 \times 10^7$  backpropagations, the PSNR is quite constant.



Figure 5.1.The validation convergence curve of the LDCNN network and the results of BM3D and SSC-GSM.

The detailed result of algorithms is depicted in Figure. 5.2. in order to visualize the edgepreserving feature of the performed LDCNN network.



Figure 5.2. Zoomed images of the result of the denoising methods. Top left: BM3D, Top right: SSC-GSM, Bottom left: LDCNN network, Bottom right: normal-dose CT

Figure. 5.3. to Figure. 5.6. Visualize performance of sample output images for different methods to grasp more intuitive feeling. All the images are drawn randomly. The first two rows are from the thoracic CT dataset. The last row is from the CATPHAN 600 CT dataset. As can be observed, BM3D and SSC-GSM perform better on the CATPHAN 600 phantom data due to its structured pattern of the spheres. However, our proposed method still produces better images as it removes more artifacts and creates sharper edges.









Figure 5.3. Visual comparison of denoising outputs on CATPHAN 600 with different contrast. From left to right: Low-dose, Normal-dose, LDCNN, BM3D method, SSC-GSM method



Figure 5.4.Visual comparison of denoising outputs with on CATPHAN 600different line spacing. From left to right: Low-dose, Normal-dose, LDCNN, BM3D method, SSC-GSM method









Figure 5.5. Visual comparison of denoising outputs on the Thoracic dataset. From left to right: Lowdose, Normal-dose, LDCNN, BM3D method, SSC-GSM method









Figure 5.6.Visual comparison of denoising outputs on the Thoracic dataset. From left to right: Lowdose, Normal-dose, LDCNN, BM3D method, SSC-GSM method

# 5.2. Deep Convolutional approach for Low-Dose CT Image Noise Reduction (Deep-LDCNN)

The presented approach was evaluated by medium-dose CATPHAN 600, piglet and thoracic datasets. Also the outcome is quantitatively and qualitatively compared to the widely known and state of the arts methods in image denoising BM3D [19], Simultaneous Sparse Coding (SSC-GSM), Learning to Separate Morphological Diversity (TV-MCA) [20], Chen CNN[36], and the previous research on CT image denoising (LDCNN).

# 5.2.1. CT Datasets

#### 5.2.2.1. Simulated Dataset

For creating the simulated dataset, we use normal-dose CT images taken from The Cancer Imaging Archive (TCIA), CATPHAN 600 and the piglet datasets. All the images have the size of  $512 \times 512$ .

The sonograms were simulated from the normal-dose CT images using fan-beam geometry. Since it was studied that the noise distribution in CT sonograms containing the Poisson noise, Poisson noise were added to the normal-dose images in order to create the corresponding low-dose CT images. As a result, the distribution can be considered as:

$$Image \sim Poisson (I_o e^{-r}) \tag{5.4}$$

Which the noise levels can be controlled by the factor  $I_o$  widely known as blank flux and r is the sinogram simulation factor.

#### 5.2.2.2. Clinical Dataset

A CT image dataset, was acquired from piglet with a variety of doses by GE scanner (Discovery CT750 HD), 906 normal-dose and the corresponding low-dose CT images with the slice thickness of 0.625 mm. CT images with 300 mA correspond to the normal dose while 5%, 10%, and 25% can be considered as the current reductions images, low-dose CT. A sequence of tube currents was utilized in the scanning process in order to provide CT images with various dose

levels. All the images have the size of 512×512. The detail of the piglet dataset is illustrated in Table 5.3.

In order to certify the performance of the proposed architecture, a widely used CT phantom named CATPHAN600 and the anthropomorphic thoracic phantom are utilized as clinical datasts.

Dose Level	Normal	25%	10%	5%
Tube current (mA)	300	75	30	15
$CTDI_{vol} (mGy)$	30.83	7.71	3.08	1.54
DLP(mGy-cm)	943.24	235.81	94.32	47.16
Effective dose (mSv)	14.14	3.54	1.41	0.71

Table 5.3. Piglet dataset setting with four different radiation doses and 0.625 mm slice thickness.

Therefore, by using three datasets, various CT images which demonstrate clinical CT imaging applications were used. Moreover, we could experiment various noise intensities by changing the suitable factor in the simulated dataset and validate the performed network. A number of typical CT images which were studied in this research are in Figure. 5.7.

As a means to provide large-scale datasets to satisfy the deep convolutional architectures and fathoming the limitation of the medical datasets mainly concerning the privacy of thoracic phantoms, image-patches extracted from this dataset in order to expand the CT image dataset. The sub-images were taken with the size of  $33 \times 33$  with the stride of 11. Moreover, training, validation and testing images were taken randomly shuffled from the normal-dose and the corresponding low-dose CT images by 50%, 25%, and 25% respectively. Furthermore, the data augmentation was used in order to double the thoracic images with the scale transformation.



Figure 5.7. Samples of studied normal-dose CT image datasets in Deep-LDCNN. 5.2.2. Evaluation Metrics and Quality Assessment Models

In this study, a number of recently established image evaluation metrics were employed. Image quality is calculated through different quality assessment models in order to compare the output of the proposed network F(Y) to the normal-dose image (X). The established peak signalto-noise ratio (PSNR), structural similarity (SSIM), root mean squared error (RMSE), universal quality index (UQI), multi-scale SSIM index (MSSIM), weighted signal-to-noise ratio (WSNR), visual information fidelity (VIF), noise quality measure (NQM), information fidelity criterion (IFC) were applied as quantitative evaluation metrics.

#### 5.2.2.1. Multi- scale SSIM (MSSIM)

SSIM takes a whole image as one block while MSSIM is applied on image region windows and calculating the average value at the end in case of the locally non-stationary spatial noise distortions. MSSIM can be expressed as the following expression:

$$MSSIM = \frac{1}{T} \sum_{i} \sum_{j} SSIM(i, j)$$
(5.5)

Where *T* is the total number of SSIM windows. MSSIM indicated high correlation with the human visual system.

#### 5.2.2.2. Universal Quality Index (UQI)

Universal quality index (UQI) which was proposed by Bovik *et al.* [41]is a normalized approach which was presented to compare the distorted image and the reference image in terms of the luminance (mean values), contrast and correlation (standard deviations) and it is consistent with the human visual system (HVS). This approach applies to the image patches (regions) to capture image local features with sliding window algorithm and then combines them together. This metric exhibits an inconsistent and poor result when the image has small first and second order statistics.

#### 5.2.2.3. Visual Information Fidelity (VIF)

Visual information fidelity (VIF) was presented by Sheikh *et al.* [42] which is determined by the amount of the mutual (relative) information between the target (noisy) image and the reference image using the luminance component of the image and the wavelet decomposition. Specifically, this algorithm quantifies the ratio of the shared information between the clean input image and the source image, and the shared information between the noisy image and the source image. It depends on the effect of the distortion on the image pattern statistics and the amount of the relative structural information which can be extracted by the human visual perception. This method exhibits the promising results on the cross-distortion performance and resemblance to HVS.

#### 5.2.2.4. Weighted Signal-to-Noise Ratio (WSNR)

Weighted signal-to-noise ratio (WSNR) which was suggested by [43] is another recent image quality metric, which mathematically defines as the ratio of the of the average weighted signal power to the average weighted noise power. In this method, weighting is derived from a frequency response, called the contrast sensitivity function (CSF) which is a linear approximation to the HVS and higher WSNR is a resemblance to higher image quality.

#### 5.2.2.5. Noise Quality Measure (NQM)

Noise quality measure (NQM) was proposed by Venkata *et al.* [43]is based on the fluctuation in contrast sensitivity with image dimensions, spatial frequency, and distance and average local luminance based on the Peli's contrast pyramid [44]. It consists of the two main steps: first, the noise and the reference images are modeled through a nonlinear contrast pyramid pixel-wise. Second, the traditional signal-to-noise ratio of the distorted image is computed concerning the modeled target image. This algorithm doesn't depend on the orientation sensitivity which results in an improved computational complexity. Moreover, this method is highly correlated with human visual results.

#### 5.2.2.6. Information fidelity criterion (IFC)

Information fidelity criterion (IFC) was performed by Sheikh et al.[38] represents the information that could ideally be recognized by a human brain from a particular sub-band in distorted and the source images. It considers the shared information between the reference and noisy images in a specific sub-band and generalizes the result for the whole frequency bands.

The network was trained using the Caffe package and Matlab R2017a. The machine properties are Intel core i7 CPU 3.4GHz and 16GB memory and the Graphics Card is GeForce GTX 1070. Figure 5.8. indicated how by increasing the number of backpropagations, the better results can be obtained. It would be valuable to state that after the  $8 \times 10^7$  backpropagations, the PSNR is quite constant.


Figure 5.8.The result of the different algorithms on the CATPHAN 600 with different line spacing in Deep-LDCNN.

## 5.2.3. Clinical dataset

#### 5.2.3.1. CATPHAN 600

Two illustrative CT image frames from CATPHAN 600 dataset were selected to analyze the performance of the proposed architecture. The first result is depicted in Figure. 5.9. which more focused on the different spacing lines with a variety of resolutions. The second result is depicted in Figure. 5.10. which more focused on the different spheres with a wide variety of contrast. It is clear that the proposed network acquired the superior performance in noise cancellation and structure conservation. In other methodologies, BM3D, SSC-GSM, and TV-MCA introduced some noise distortion, blurred contrast regions, or low identifiable parts. However, the previous network (LDCNN) and Chen CNN provided better results but it still some blurred lesions remained. Furthermore, the proposed network could preserve the detailed edges and corners. It can be due to the small feature map sizes in the proposed algorithm compared to Chen CNN and the previous algorithm. The convolutional filters, especially in the last layer, due to the averaging characteristics of this mathematical method, introduces blurred edges and by increasing the filter map sizes, this blurring artifact elevates. In the proposed network, the feature map sizes in the last layer are smaller than the other studied CNNs and as a result, it yields better results.

Table 5.4. summarized the performance of the utilized mythologies. It is obvious that the proposed network achieved superior performance regarding the evaluation metrics. Stating that the network result was based on the  $8 \times 10^7$  backpropagations. By providing signal-based evaluations and visually-pleasing quality metrics, the proposed method exhibits a promising performance. It is also worth stating that due to the spherical structures of the CATPHAN 600 dataset, which is a resemblance to the Gaussian filters, all the methods provide an excessive noise suppression and can acquire higher PSNR among all the other datasets.

Eval. Mat	Deep-LDCNN	LDCNN	BM3D	SSC-GSM	TV-MCA	Chen CNN
PSNR (dB)	43.2701	41.4976	38.9322	37.9603	38.1115	41.403
RMSE	0.0069	0.0084	0.0113	0.0126	0.0124	0.0085
SSIM	0.9524	0.9471	0.9297	0.9305	0.9305	0.944
MSSIM	0.9561	0.9421	0.9278	0.9381	0.9381	0.9445
UQI	0.8064	0.7694	0.6076	0.6504	0.6703	0.8034
WSNR (dB)	46.785	44.354	41.3130	32.1156	32.4769	45.589
VIF	0.9780	0.954	0.8894	0.8545	0.8616	0.9712
NQM	41.564	35.563	29.1657	28.8546	28.6544	36.6134
IFC	5.2347	5.167	2.1757	2.1915	2.2047	5.154

Table 5.4. The average results of quality metrics on the CATPHAN 600 in Deep-LDCNN performance.



Figure 5.9. The result of the different algorithms on the CATPHAN 600 with different line spacing in Deep- LDCNN performance. From left to right: Low-dose, Normal-dose, Deep- LDCNN, BM3D method, SSC-GSM method, TV-MCA method, LDCNN, Chen CNN.



Figure 5.10. The result of the different algorithms on the CATPHAN 600 with different contrast in Deep- LDCNN performance. From left to right: Low-dose, Normal-dose, Deep- LDCNN, BM3D method, SSC-GSM method, TV-MCA method, LDCNN, Chen CNN.

#### 5.2.3.2. Piglet Dataset

Two sample CT images are depicted in Figure. 5.11. and Figure. 5.12. in order to validate the performance. Note that the images are drawn randomly to represent two distinctive regions of the piglet. As it is clear, the proposed network can significantly improve the low-dose CT image (artifact reduction) and keep the inner structures and edges and fine details, while other methodologies, especially BM3D, SSC-GSM, and TV-MCA still seem noisy.

Table 5.5. provides the comparison of the different algorithms on piglet dataset. The proposed network achieved the best performance among all the algorithms.

Eval. Mat	Deep- LDCNN	LDCNN	BM3D	SSC-GSM	TV-MCA	Chen CNN
PSNR (dB)	42.2701	40.024	34.3748	36.9603	37.546	40.001
RMSE	0.0077	0.01	0.0191	0.0142	0.0133	0.01
SSIM	0.943	0.931	0.8856	0.8967	0.897	0.928
MSSIM	0.941	0.9345	0.8876	0.8976	0.8978	0.9267
UQI	0.7834	0.7545	0.6734	0.7044	0.7344	0.7411
WSNR (dB)	42.4532	40.9454	35.6422	37.3912	38.4367	42.4912
VIF	0.978	0.9611	0.8959	0.9012	0.9011	0.9744
NQM	27.8868	25.8945	23.4582	24.4581	24.9845	27.1292
IFC	3.4832	3.4191	2.8745	2.9788	3.0912	3.1245

Table 5.5. The average results of quality metrics on the Piglet dataset in Deep- LDCNN performance.



Figure 5.11. The result of the different algorithms on the Piglet dataset in Deep- LDCNN performance. From left to right: Low-dose, Normal-dose, Deep- LDCNN, BM3D method, SSC-GSM method, TV-MCA method, LDCNN, Chen CNN



Figure 5.12. The result of the different algorithms on the Piglet dataset in Deep- LDCNN performance. From left to right: Low-dose, Normal-dose, Deep- LDCNN, BM3D method, SSC-GSM method, TV-MCA method, LDCNN, Chen CNN .

#### **5.2.3.3.** Thoracic Phantom Dataset

Methodologies are compared on the anthropomorphic thoracic phantom dataset and Figure. 5.13. and Figure. 5.14. ate two random CT images to be the representatives of the dataset. Figure. 5.13. concentrates on the outer lung regions and Figure. 5.14. focuses on the inner lung and small structures (mainly consists of blood vessels). Notwithstanding the fact that all the convolutional neural network approaches are successful in preserving the sharp edges and other algorithms cannot provide a reliable contrast robustness and artifact suppression, the proposed Deep-LDCNN method can significantly remove the noise within the lung which is a significant achievement.

Table 5.6. summarize the quality metrics, and it was clear that in all the visual-based assessments (such as SSIM, UQI, and VIF) and signal-based metrics (PSNR, and RMSE), the Deep-LDCNN approach can attain the highest performance.

Eval. Mat	Deep-LDCNN	LDCNN	BM3D	SSC-GSM	TV-MCA	Chen CNN
PSNR (dB)	33.3452	31.456	27.4814	26.8704	27.0391	31.241
RMSE	0.0215	0.0267	0.0423	0.0453	0.0445	0.0274
SSIM	0.9624	0.9544	0.9044	0.9102	0.9134	0.9599
MSSIM	0.9641	0.9599	0.9023	0.92	0.9201	0.9611
UQI	0.8912	0.8811	0.8411	0.8478	0.8522	0.8823
WSNR (dB)	34.8473	34.123	30.583	31.005	31.194	34.948
VIF	0.9834	0.9832	0.8034	0.8845	0.9022	0.9782
NQM	29.453	28.124	25.234	26.983	26.999	28.934
IFC	3.857	3.276	2.987	2.991	3.002	3.713

 Table 5.6. The average results of quality metrics on the thoracic phantom dataset in Deep- LDCNN performance.



Figure 5.13. The result of the different algorithms on the thoracic phantom dataset in Deep- LDCNN performance. From left to right: Low-dose, Normal-dose, Deep- LDCNN, BM3D method, SSC-GSM method, TV-MCA method, LDCNN, Chen CNN .



Figure 5.14. The result of the different algorithms on the thoracic phantom dataset in Deep- LDCNN performance. From left to right: Low-dose, Normal-dose, Deep- LDCNN, BM3D method, SSC-GSM method, TV-MCA method, LDCNN, Chen CNN .

### 5.2.4. Simulated dataset

To validate the result of the proposed Deep-LDCNN network, artificial datasets are created. First, the CT sinogram is computed from normal-dose CT images by imposing a fan-beam methodology. Then, a low-dose CT dataset was generated by applying Poisson noise to the sinogram. The advantage of this method is, the noise power can be controlled by adjusting a noise factor.

Figure. 5.15, 5.16, and 5.17. are sample images for each dataset. In this approach, first four types of low-dose CT images are created by applying Poisson noise with different levels of the noise factor  $I_0 = 10^4$ ,  $3 \times 10^4$ ,  $5 \times 10^4$ ,  $10^5$ . The noisiest images  $I_0 = 10^5$ , which are suffer from the critical deterioration and most of the structures cannot be discerned. The random combination of all the low-dose levels (200 images from each set) are considered as a low-dose CT, and along with the normal-dose CT constitute a training set. Moreover, the noisy images  $I_0 = 10^4$ ,  $3 \times 10^4$ ,  $5 \times 10^4$ ,  $10^5$  are considered as a testing and validation sets. In this way, not only the stimulated dataset was evaluated which is the ultimate aim, but also the generalization characteristic of the network is also examined. Since, one of the issues in CT denoising, is that the low-dose CT can be obtained by adjusting all the radiation doses, which are lower than normal-dose radiation. The proposed Deep-LDCNN can achieve the promising performance by preserving the detailed structures and being more pleasant for visual inspection.













Figure 5.15. A sample image from the piglet dataset imposed on the simulated low dose and the result of the Deep- LDCNN. The first row is the normal full dose. The left column is the created low-dose with  $I_0 =$  $10^4$ ,  $3 \times 10^4$ ,  $5 \times 10^4$ ,  $10^5$  respectively and the right column is the output of the Deep- LDCNN on the corresponding CT image.













Figure 5.16. A sample image from the thoracic dataset imposed on the simulated low dose and the result of the Deep- LDCNN. The first row is the normal full dose. The left column is the created low-dose with  $I_0 = 10^4$ ,  $3 \times 10^4$ ,  $5 \times 10^4$ ,  $10^5$  respectively and the right column is the output of the Deep- LDCNN on the corresponding CT image.

















Figure 5.17. A sample image from the CATPHAN 600 dataset imposed on the simulated low dose and the result of the Deep- LDCNN. The first row is the normal full dose. The left column is the created low-dose with  $I_0 = 10^4$ ,  $3 \times 10^4$ ,  $5 \times 10^4$ ,  $10^5$  respectively and the right column is the output of the Deep- LDCNN on the corresponding CT image.

### 5.2.5. Architecture Robustness

In this part, the concentrate is on the fact that the piglet dataset, consists of four different radiation doses (300, 75, 30, *and* 15 mA), which each of the last three radiation currents can be considered as a low-dose CT dataset. Generally speaking, low-dose CT can be any radiation currents which is lower than the normal-dose CT image. Moreover, the medical convolutional neural networks are trained based on the specific low-dose and cannot satisfactory perform on different noise levels. As a result, we randomly select the low-dose CT images by different noise levels from the piglet dataset (600 images from 2 different noise levels, 15 and 30) along with the corresponding normal-dose CT images to from a training set. A sample test image is depicted in Figure. 5.18. to evaluate the result and it proves the performance robustness of the proposed Deep-LDCNN architecture.















Figure 5.18. A sample image from the piglet dataset with different radiation dose and the result of the Deep- LDCNN. The first row is the normal full dose. The left column is the clinically generated low-dose with the dosage 25%, 10%, 5% respectively and the right column is the output of the Deep- LDCNN on the corresponding CT image.

### 5.2.6. Computational Cost and Running Time

In general, deep learning approaches are possessing the computational cost over the dictionary learning and iterative statistical methodologies. By introducing high-level GPUs, the training time of convolutional neural networks is enhanced. The BM3D method took one minute for denoising each CT image. The SSC-GSM method took five minutes to be implemented on each image since it exploits a complicated structure to estimate the noise distortions. The Chen CNN, which is comprised of three layers, took ten hours to train 10<sup>6</sup> patches in *Caffe* framework based on the original configuration. Afterwards, it took twenty seconds to test on each CT image in the testing set. The LDCNN and Deep-LDCNN took twelve and nine hours respectively to train 10<sup>6</sup> patches and the testing time is about forty and eleven seconds respectively as the Deep-LDCNN is comprised of about 18,000 parameters while LDCNN consists of 256,000 parameters. Note that the training time was calculated until the RMSE of the network reach to 0.01. As a result, the deep learning proposed approaches exhibit more efficient performance than other methods regarding the computational cost and running time.

# Chapter 6 Conclusion and Future Works

# 6.1. Convolutional Architecture for Low-Dose CT Noise Reduction (LDCNN)

In this thesis, a fully convolutional framework is presented in order to learn an end-to-end mapping between low-dose and normal-dose CT images. It consists of four convolutional layers with ReLU as an activation function. All the network parameters are trained by backpropagation technique in order to obtain the optional values. The network structure (number of layers, size of feature maps, and the number of filters) is investigated in detail. The performed network can be considered as the combination of the dictionary learning methods, which was the prevailing

methods before deep neural networks, and deep learning frameworks. For each convolutional layer, the corresponding function in dictionary learning framework is explained and the dictionary learning steps are formulated as the convolutions in the hidden layers. The performance was superior to the dictionary learning algorithms in terms of achieving the high-quality CT images. The results of the low-dose CT architecture are evaluated by two CT phantom datasets and indicate that the proposed architecture is capable of detecting low-contrast objects.

# 6.2. Deep Convolutional approach for Low-Dose CT Image Noise Reduction (Deep-LDCNN)

The deep convolutional network is introduced to enhance the training and the performance of the LDCNN. The primary element, which makes the Deep-LDCNN better than the LDCNN is using multiple convolutional layers in a mapping stage, namely compressing layer, mapping layers, and expanding layers. Moreover, the development process can be considered as fine-tuning a convolutional network, which is very popular in medical applications. The performance of the network is compared with three sparse coding methods along with the Chen CNN, which is a recently published paper in low-dose CT denoising. It was demonstrated that the proposed algorithm has an outstanding merit due to its simplicity and superiority of performance compared to the most commonly used methods. The results of the low-dose CT network indicate that the proposed architecture is capable of efficiently denoising low-dose CT images. Due to the presented advantages, the proposed structure could be adapted in various computer vision applications such as image inpainting and image interpolation.

In this study, three CT datasets are used to evaluate the results, which each dataset has distinct features. In the thoracic dataset, the network is evaluated to detect small objects. In CATPHAN 600 phantom, the focus is on the different line spacing and contrast. Furthermore, the architecture is evaluated using the piglet dataset. The simulated CT datasets are utilized with different noise level in order to demonstrate the robustness of the Deep-LDCNN features. The proposed network architecture support input CT images of any arbitrary sizes with variable X-ray doses. The network is capable of denoising a variety of radiation dosages.

### 6.3. Future improvements

As the deep learning is rapidly growing and its performance keep surpassing the previous methods, further achievements are gaining, and the deep learning becomes the interest of the majority of the researchers in all the fields of science. In order to improve the network structure, employing the residual frameworks is suggested. Moreover, utilizing the fine-tuning technique is advantageous due to improving the computational complexity of the deep learning approaches. Utilizing generative adversarial networks and block-matching CNNs can also be useful, which are the recent and updated convolutional frameworks.

## References

- D. Brenner and E. Hall, "Computed tomography-an increasing source of radiation exposure", *N. Engl. J. Med.*, vol. 357, no. 22, pp. 2277–2284, 2007.
- [2] S. Berrington de Gonzalez, A. Darby, "Risk of Cancer from Diagnostic x-rays: Estimated for the UK and 14 Other Countries", *Lancet*, vol. 363, no. 9406, pp. 354–351, 2004.
- J. Mathews *et al.* "Cancer risk in 680 000 people exposed to computed tomography scans in childhood or adolescence: data linkage study of 11 million Australians", *BMJ*, vol. 346, pp. 23-60, May 2013.
- [4] C. Martin, D. Sutton, and P. Sharp, "Balancing patient dose and image quality", *Appl. Radiat. Isot.*, vol. 50, no. 1, pp. 1–19, 1999.
- [5] A. Kak and M. Slaney, "Principles of Computerized Tomographic Imaging", *IEEE Press*, vol. 4, no. 167, pp. 1–5, 1988.
- [6] L. Shepp and B. Logan, "Reconstructing interior head tissue from x-ray transmissions", *IEEE Trans. Nucl. Sci.*, vol. 21, no. 1, pp. 228–236, 1974.
- Y. Wang, J. Yang, W. Yin, and Y. Zhang, "A New Alternating Minimization Algorithm for Total Variation Image Reconstruction", *SIAM J. Imaging Sci.*, vol. 1, no. 3, pp. 248–272, 2008.
- [8] L. Goldman, "Principles of CT: radiation dose and image quality", *J Nucl. Med. Technol.*, vol. 35, no. 4, pp. 213-228, 2007.
- [9] J. Hsieh, R. Molthen, C. Dawson, and R. H. Johnson, "An iterative approach to the beam hardening correction in cone beam CT", *Med. Phys.*, vol. 27, no. 1, pp. 23–29, 2000.
- [10] C. Yan, R. Whalen, G. Beaupré, S. Yen, and S. Napel, "Reconstruction algorithm for polychromatic CT imaging: application to beam hardening correction", *IEEE Trans. Med. Imaging*, vol. 19, no. 1,

pp. 1–11, 2000.

- P. La Rivière, "Penalized-likelihood sinogram smoothing for low-dose CT", *Med. Phys.*, vol. 32, no. 61, pp. 1676–1683, May 2005.
- [12] M. Kachelrieß, O. Watzke, and W. A. Kalender, "Generalized multi-dimensional adaptive filtering for conventional and spiral single-slice, multi-slice, and cone-beam CT", *Med. Phys.*, vol. 28, no. 4, pp. 475–490, 2001.
- [13] A. Khodabandeh, J. Alirezaie, P. Babyn, and A. Ahmadian, "Computed Tomography Image Denoising by Learning to Separate Morphological Diversity", *Telecommun. Signal Process. (TSP)*, 2015 38th Int. Conf., pp. 513–517, 2015.
- [14] A. Chambolle, "An algorithm for total variation minimizations and applications", *J. Math. Imaging Vis.*, vol. 10, no. 1–2, pp. 89–97, 2004.
- [15] S. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression", *IEEE Trans. Image Process.*, vol. 9, no. 9, pp. 1532–1546, 2000.
- [16] J. Thibault, K. Sauer, C. Bouman, and J. Hsieh, "A three-dimensional statistical approach to improved image quality for multislice helical CT", *Med. Phys.*, vol. 34, no. 11, pp. 4526–4544, 2007.
- [17] D. Fleischmann and F. Boas, "Computed tomography-old ideas and new technology", *Eur. Radiol.*, vol. 21, no. 3, pp. 510–517, 2011.
- [18] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition", *Proc. 27th Asilomar Conf. Signals, Syst. Comput.*, pp. 1–5, 1993.
- [19] L. Chen, S. P. Gou, Y. Yao, J. Bai, and L. Jiao, "Denoising of Low Dose CT Image with Context-Based BM3D", IEEE Region 10 Conference (TENCON), Singapore, pp. 682-685 2016.

- [20] W. Dong, G. Shi, Y. Ma, and X. Li, "Image Restoration via Simultaneous Sparse Coding: Where Structured Sparsity Meets Gaussian Scale Mixture", *Int. J. Comput. Vis.*, pp. 1–16, 2015.
- [21] Y. Cun, J. Denker, D. Henderson, R. Howard, W. Hubbard, "Handwritten digit recognition with a backpropagation network", Advances in neural information processing systems, pp. 396-404 1990.
- [22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", *NIPS'12 Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, pp. 1097–1105, 2012.
- [23] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", Int. Conf. Learn. Represent., pp. 1–14, 2015.
- [24] C. Szegedy et al., "Going deeper with convolutions", 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–9, 2015.
- [25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", arXiv Prepr. arXiv1512.03385v1.
- [26] Y. LeCun, L. Bottou, G. Orr, "Efficient backprop", Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 7700 LECTU, pp. 9–48, 2012.
- [27] V. Nair and G. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines", Proc.
   27th Int. Conf. Mach. Learn., no. 3, pp. 807–814, 2010.
- [28] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification", 2009 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, pp. 1794– 1801, 2009.
- [29] M. Yadong, L. Wei, F. Wei, "Stochastic Gradient Made Stable : A Manifold Propagation Approach for Large-Scale Optimization", arXiv:1506.08350v2.
- [30] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors", *ArXiv e-prints*, pp. 1–18, 2012.

- [31] S. Hauberg, O. Freifeld, A. Larsen, J. Fisher, and L. Hansen, "Dreaming More Data: Classdependent Distributions over Diffeomorphisms for Learned Data Augmentation", *Artif. Intell. Stat.*, vol. 51, pp. 342–350, 2016.
- [32] A. Choromanska, M. Henaff, M. Mathieu, G. Ben Arous, and Y. LeCun, "The Loss Surfaces of Multilayer Networks", *Aistats*, vol. 38, pp. 192-204, 2015.
- [33] J. Antony, K. McGuinness, N. Connor, and K. Moran, "Quantifying Radiographic Knee Osteoarthritis Severity using Deep Convolutional Neural Networks", arXiv:1609.02469v1
- [34] H. Suk, S. Lee, and D. Shen, "Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis", *Neuroimage*, vol. 101, pp. 569–582, 2014.
- [35] A. Arindra *et al.*, "Pulmonary Nodule Detection in CT Images: False Positive Reduction Using Multi-View Convolutional Networks", *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1160–1169, 2016.
- [36] H. Chen, Y. Zhang, W. Zhang, P. Liao, K. Li, and J. Zhou, "LOW-DOSE CT DENOISING WITH CONVOLUTIONAL NEURAL NETWORK" arXiv:1610.00321v1.
- [37] E. Kang, J. Min, and J. Ye, "Wavelet Domain Residual Network (WavResNet) for Low-Dose X-ray CT Reconstruction", arXiv:1703.01383v1.
- [38] Z. Wang, C. Bovik, H. Sheikh, and E. Simmoncelli, "Image quality assessment: form error visibility to structural similarity", *Image Process. IEEE Trans.*, vol. 13, no. 4, pp. 600–612, 2004.
- [39] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical Evaluation of Rectified Activations in Convolution Network", *ICML Deep Learn. Work.*, pp. 1–5, 2015.
- [40] M. Gavrielides *et al.*, "A resource for the assessment of lung nodule size estimation methods: database of thoracic CT scans of an anthropomorphic phantom", *Opt Express*, vol. 18, no. 14, pp. 15244–15255, 2010.

- [41] H. Sheikh, A. Bovik, and G. Veciana, "An information fidelity criterion for image quality assessment using natural scene statistics", *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2117– 2128, 2005.
- [42] H. Sheikh and A. Bovik, "Image information and visual quality", *IEEE Trans. Image Process.*, vol. 15, no. 2, pp. 430–444, 2006.
- [43] N. Damera-Venkata, T. Kite, W. Geisler, B. Evans, C. Bovik, "Image quality assessment based on a degradation model", *IEEE Trans. Image Process.*, vol. 9, no. 4, pp. 636–650, 2000.
- [44] J. Johnson, E. Krupinski, M. Yan, H. Roehrig, A. Graham, and R. Weinstein, "Using a Visual Discrimination Model for the Detection of Compression Artifacts in Virtual Pathology Images", *IEEE Trans. Med. Imaging*, vol. 30, no. 2, pp. 1011-1025, 2011.