DEEP LEARNING FOR LOW-DOSE CT NOISE REMOVAL USING DILATED CONVOLUTION AND PERCEPTUAL LOSS

by

Maryam Gholizadeh-Ansari

Master of Applied Science, Sharif University of Technology, 2001
Bachelor of Engineering, Tehran University, 1998

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, 2019

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 dissertation may be made electronically available to the public.

DEEP LEARNING FOR LOW-DOSE CT NOISE REMOVAL USING DILATED CONVOLUTION AND PERCEPTUAL LOSS

Master of Applied Science 2019

Maryam Gholizadeh-Ansari

Electrical and Computer Engineering

Ryerson University

Abstract

Low-dose computed tomography has been recommended to reduce the radiation risks of CT scans for patients. However, the reconstructed CT image will be considerably degraded because of photon starvation. Both traditional noise removal techniques and neural networks have been used to enhance the quality of low-dose CT images. In this study, a deep neural network is proposed to mitigate this problem. The network employs dilated convolution, batch normalization, and residual learning. Moreover, a non-trainable edge detection layer is proposed helping to produce sharper edges in the output image without introducing additional complexity. This network is optimized by a combination of mean-square error and perceptual loss to preserve textural details in the CT image that are critical for diagnosis. This objective function solves the over-smoothing problem and grid-like artifacts caused by per-pixel loss and perceptual loss, respectively. The experiments demonstrate the effects of each modification to the network and confirm that the proposed network achieves better performance relative to the state of the art methods.

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 family 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

List of Tables

List of Figures

Acronyms

Adam Adaptive Momentum Estimation.

ART Algebraic Reconstruction Technique.

BM3D Block Matching and 3D Filtering.

BN Batch Normalization.

CNN Convolutional Neural Network.

CT Computed Tomography.

DA Denoising Autoencoders.

DCT Discrete Cosine Transform.

DFT Discrete Fourier Transform.

DICOM Digital Imaging and Communications in Medicine.

DRL Dilated Residual Learning.

DRL-E Dilated Residual Learning with Edge Detection Layer.

DRL-E-M DRL-E Network Optimized by MSE.

DRL-E-MP DRL-E Network Optimized by MSE and Perceptual loss.

DRL-E-P DRL-E Network Optimized by Perceptual Loss.

DSSIM Structural Dissimilarity.

FBP Filtered Back Projection.

FFT Fast Fourier Transform.

GAN Generative Adversarial Networks.

HU Hounsfield Unit.

K-SVD K-Singular Value Decomposition.

KL Kullback-Leibler.

MBIR Model-Based Iterative Reconstruction.

MOD Method of Optimal Direction.

MSE Mean Square Error.

NEC Noise-Equivalent Count.

NLP Natural Language Processing.

NP Nondeterministic Polynomial.

OS-SIRT Simultaneous Iterative Reconstruction Technique.

PCA Principal Component Analysis.

PSNR Peak Signal to Noise Ratio.

ReLU Rectified Linear Unit.

RF Receptive Field.

SAFIRE Sinogram Affirmed Iterative Reconstruction.

SART Simultaneous Algebraic Reconstruction Technique.

SGD Stochastic Gradient Descent.

SNR Signal to Noise Ratio.

SSIM Structural Similarity.

TCIA The Cancer Imaging Archive.

Chapter 1

Introduction

1.1 Background

X-ray computed tomography (CT) is a technique to create images from inside the body by employing X-ray and a computer. It generates pictures that are more detailed compared to a conventional X-ray image. One session of CT scan produces a series of cross-sectional images, also called slices that provide a view of the inner organs. It is a non-invasive method to view inside the patient's body and detect abnormalities without performing surgeries. CT has been used widely in many countries around the world. The industry has also adopted this method to measure volumetric information in 3-dimensional for an object without the need to destruct it. However, the main application of CT scan is medical imaging to make a diagnosis.

1.2 Problem Statement

The invention of Computed Tomography in 1972 has presented a great help for physicians to determine diseases such as cancer. However, studies have shown that multiple CT scans may cause cancer, too. The number of CT imaging in the past decade have raised dramatically. In 2015, over 80 million scans were performed in the United States. It is estimated that around 1.5% to 2% of all cancers in 2007 are the result of previous CT scans [1]. Therefore, it is necessary to limit the radiation risk for the patients. Low-dose Computed Tomography (CT) is considered a solution to restrict a patient's X-ray

exposure during a CT scan session.

Reducing the X-ray current lowers the amount of radiation for the patient. However, it generates images that are significantly degraded by noise and artifacts, so the reconstructed picture may not be reliable for making a diagnosis. This need has made noise removal from low-dose CT images an active research area. Photon starvation because of low-dose current generates noise that can be modeled by Poisson noise in the projection domain. However, transforming the data to image space changes the nature of the noise entirely. In this domain, the characteristics of the noise are unknown which makes the noise removal task very challenging. Keeping the details in a CT image is critical while many traditional denoising tasks blur or over-smooth these details that make them inappropriate for the job.

Doinsing algorithms of CT images are divided into three groups: Denoising the sinogram data in the projection space, denoising the reconstructed image, and iterative methods that operate in both domains. The proposed algorithm in this study removes noise from the reconstructed Ct image.

1.3 Methods

Deep learning is a branch from the machine learning family, and the goal is to learn the representation of the data. During the training process, the algorithm observes many samples and finds the distribution of the target with respect to the input data. It is specially beneficial in identifying the patterns of unstructured data, similar to the problem of this study. Deep learning has advanced significantly in recent years and provided successful results in different fields such as image processing, audio signal processing, text processing, and business analytics. Stacking more layers in neural networks and developing techniques to train and improve the performance has lead to deep learning. Convolutional layers with sparse connections have replaced the fully connected layers in most of the networks. Techniques such as batch normalization, residual learning, and use of different loss function have boosted the outcomes.

In this study, deep learning is utilized to remove noise from low-dose CT images. It is shown that the proposed deep network is capable of generating normal-dose from low-dose one. The proposed network outperforms state of the art BM3D [2] which does

not use neural networks by a large margin.

1.4 Framework

In this study, two deep networks are proposed. The first network called Dilated Residual Learning is a seven-layer network that employs dilated convolution instead of standard convolution. Dilated convolution helps to achieve better results in the fewer layer. The network also takes advantage of residual learning and batch normalization to improve the performance.

The second network is the advancement of the first network that uses the proposed edge detection layer and a combination of per-pixel loss and perceptual loss. This network achieves sharper edges and is capable of preserving the textural structure in the CT image. This performance is obtained with minimal change in the number of weights. It is possible to further improve the outcome by stacking more layer; however, we have researched techniques to perform denoising efficiently and without increasing the complexity of the network.

One of the challenges in deep learning, especially in the medical field is that not enough training data is available. Training a deep network similar to other machine learning algorithms needs a lot of data. The higher number of data leads to better characterization and therefore, more accurate results. To address the problem, we have generated simulated low-dose CT images. To evaluate the performances of the networks this dataset is used as well as a real and a phantom CT dataset.

1.5 Contributions

Deep convolutional networks have been used previously to remove noise from low-dose CT images. However, this study is the first that uses dilated convolution to perform this task. The first proposed network combines multiple techniques to accomplish good performance. Symmetric shortcut connections are used to take full advantage of residual learning and push the performance further.It also does not suffer from over-smoothing which is the caused of optimizing by per-pixel losses and check-board artifacts that is caused by perceptual loss.

We have also proposed an edge detection layer that emphasizes the edges in the image and allows to generate output images with sharper boundaries. The proposed layer does not have any trainable weights and therefore do not introduce any complications. Furthermore, our experiments have shown that mean square error is not the most suitable objective function for optimizing the networks. It generates blurred output images and fails to recover many textural details. To resolve this problem, an objective function is proposed that joins both mean square loss and perceptual loss. This objective function takes advantage of the benefits offered by each loss function. It also does not suffer from over-smoothing which is the caused by per-pixel losses and check-board artifacts that is caused by perceptual loss.

Based on this research, a conference paper is published in the proceedings of the 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC2018) [3]. In addition, a journal manuscript is submitted to the Journal of Digital Imaging, Springer and a conference paper is also submitted to the 41th Annual International Conference of the IEEE EMBC2019.

1.6 Overview of the Thesis

This thesis is organized as follows. Chapter 2 gives an introduction to the basics of X-ray computed tomography and image reconstruction methods. The risks of performing multiple CT scans and the motivation behind this study is explained, as well. The chapter gives literature review of the previous studies about removing noise from low-dose CT images. Chapter 3 illustrates the principles of deep learning including optimization algorithms, convolutional layers, and common network structures. In chapter 4, the architectures of the proposed networks are explained through the analysis of each component and the reason for employing it. Chapter 5 provides information about preprocessing CT images and dataset preparations. The results of denoising three CT datasets are provided in chapter 6. In this section, multiple experiments are performed to demonstrate how each component contributes to removing noise from low-dose CT. Finally, chapter 7 provides a summary and avenues for possible future research studies.

Chapter 2

Computed Tomography

2.1 CT Scan Fundamentals

Computed Tomography (CT) is a diagnostic tool used to create images of the internal organs and bones of the human body. It is used by clinicians to detect suspicion of health conditions such as tumours, bone fractures, and vascular diseases within the region of interest. CT scanners generate images of the body by shaping x-ray beams through a collimator that is passed through the patient and is received by detector arrays. The x-rays diverge in a fan-beam pattern to the detector. The gantry rotates the detector arrays and x-ray beam around the patient to capture the region of interest at different angles to create slices. In Figure 2.1, the gantry rotates around the patient to construct slices of the region of interest.

The collected photons are then converted to an electric signal using a photodiode. The data is recorded as the linear attenuation coefficients, which is the reduction in x-ray intensity due to light absorption from organs. The attenuation affects the intensity of the photons I collected by the detector array.

$$I = I_o e^{-\mu x} \tag{2.1}$$

Here, I_0 represents the incident flux. The linear attenuation μ is converted into Hounsfield Unit (HU) using Equation 2.2, which is a reference defining attenuation value with respect

to water [5].

$$HU = 1000 - \frac{\mu - \mu_{water}}{\mu_{water}} \tag{2.2}$$

The Hounsfield scale is used to classify the observed tissue and can indicate clots and tumours based on variance from healthy HU values for the specified organs.

Based on the intensity of the photons that the detector array receives, the pixels are assigned values that will map out the image gradient. Table 2.1 displays the Hounsfield values for different tissues. When the tissue is dense such as teeth or bone, it absorbs most of the X-ray. Therefore, fewer X-ray beams are collected by the detector and that part will be seen lighter in the CT scan image. On the other hand, organs such as lung that is full of air are shown darker in the image. To obtain the CT image, it undergoes a process called image reconstruction.

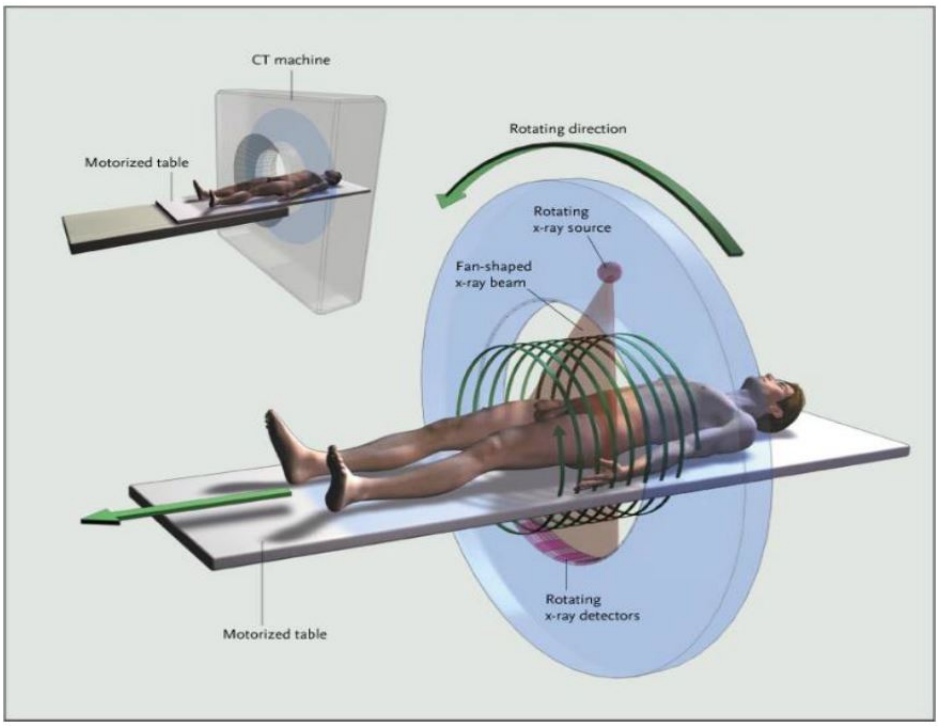


Figure 2.1: CT Scanner Acquisition [4]

Table 2.1: Hounsfield units with grayscale demonstration

Tissue	**Hounsfield Units (HU)**
Air	-1000
Lung	-400 to -600
Fat	-60 to -100
Water	0
Muscle	10 to 40
Blood	30 to 45
Bone	700 to 3000

2.2 CT Image Reconstruction

To generate a CT image, some pre-processing techniques are used in image reconstruction. The Radon transform is an important part of image preprocessing that converts an image to line integrals along different angles [4]. The result is a sinogram, which consists of the projections at a different angle and has a characteristic sinusoid pattern that repeats every 180 degrees [4]. Processing techniques are applied to the sinogram using inverse radon transform [4].

2.2.1 Filtered Back Projection

Filtered Back Projection (FBP) is one of the first techniques developed for image pre-processing. FBP is based on the Fourier slice theorem, which states that the Fourier transform of the projection can be equated to its Fourier transform along a radial line [4]. The slice can be created by taking the Fourier slice theorem over enough angles to reconstruct the image [4]. FBP applies Fourier slice theorem by multiplying the Fourier transform projection by a width of $2\pi\omega/k$ to give an approximation mapped out across the whole frequency domain [4].

FBP can be computed using either convolution or a transfer function. Discrete Fourier Transform (DFT) is applied over a specified number of discrete points then periodic convolution is used with a ramp function such as Hamming window to filter high-frequency dose [4, 6]. Finally, inverse Fourier transform is applied to return the signal back to the time domain [4]. Fast Fourier Transform (FFT) is used in computation over DFT due

to its quicker processing speed. If the transfer function method is used, then it involves using the impulse function in the frequency domain and is digitized over the sampling interval [6]. Using the sampling interval, the Nyquist theory is applied to state that the sampling frequency must be twice the maximum signal frequency to prevent aliasing [4]. Zero padding is a technique that adds zeroes to the end of the signal in the time domain, which is used to improve the resolution of the signal and reduce artifacts [4].

2.2.2 Iterative and Algebraic reconstruction

Most reconstruction techniques using linear or non-linear systems present Nondeterministic Polynomial (NP) hard equations where there are more unknowns than equations. Polynomial-time algorithms are developed to solve NP-hard problems, but with increasingly complex problems it exponentially increases computation time.

Iterative and algebraic reconstruction is used more often than FBP due to its accuracy. Algebraic Reconstruction Technique (ART) creates a system of linear equations representing pixels of raw data, reconstructed voxels, and an estimate of the raw data (a matrix consisting of the line integral of the linear attenuation coefficient) [7]. The algorithm will find the error of the reconstruction then adds the correction to the image [4]. The cost function for the error is the mean squares errors[7]. The consequence of ART is the salt and pepper noise generated from using this technique [4].

Iterative reconstruction uses the process of repeating the reconstruction until it converges to the error threshold or the specified number of iterations is reached [7]. The values are only changed once all equations are computed then it undergoes the next iteration [4]. Another processing technique is Simultaneous Iterative Reconstruction Technique (OS-SIRT) where the projection data is separated into subsets instead of processing the whole projection at once [7]. The use of subsets increases the speed of convergence but could increase noise due to overcorrection [7].

Simultaneous Algebraic Reconstruction Technique (SART) uses concepts from both ART and SIRT combining the speed of ART and the error reduction of SIRT [4]. ART looks at single pixels (single x-rays) while SART looks at the whole projection to increase computation speed, but risks some noise [7]. For its correction, it uses a longitudinal Hamming window, which focuses on the center of the rays in the reconstruction circle [4]. A single iteration of SART has a low error and good quality [4].

Table 2.2: Effective dose and comparison to background radiation

Procedure	**Approximate effective radiation dose**	**Comparable to natural background radiation for**
Chest CT	7 mSv	2 years
Lung Cancer Screening CT	1.5 mSv	6 months
Chest X-ray	0.1 mSv	10 days
Abdomen and Pelvis CT	10 mSv	3 years
Abdomen and Pelvis, repeated with and without contrast material	20 mSv	7 years
Colonography	6 mSv	2 years
Head	2 mSv	8 months
Head, repeated with and without contrast material	4 mSv	16 months
Spine CT	6 mSv	2 years

2.3 Patient Health and CT Dosage

While using CT scan have been a great help for doctors to diagnose diseases, there are concerns about the risk of x-ray radiation. High levels of radiation can damage cells, which predominantly lead to concerns of developing cancer. Furthermore, the x-ray beam that passes through the body affects each organ differently. The amount of the radiation that a patient body receives is known as effective dose, and scientists measure it in millisievert (mSV). Effective noise depends on organ sensitivity to x-ray radiation and is used by doctors to evaluate the risk. In daily life, the human body is exposed to different sources of radiation such as cosmic rays from coast-to-coast round-trip airline flight and radon gas. Generally, a person who lives in the US receives about $3mSV$ per year from cosmic and natural radiation, which is known as background radiation [8]. To understand the radiation risks of a CT scan imaging, we can compare the radiation risk with the background radiation. Table 2.2 shows the effective radiation dose for some computed tomography procedures and their comparison with the background radiation.

Table 2.2 displays that the effective dose in CT scan is much higher than conventional X-ray imaging. For example, the effective dose in the chest x-ray ($0.1mSV$) is equal to

10 days of background radiation whereas chest CT exposes the body to $7mSV$, which is equal to 2 years of background radiation [8]. The current dosage of CT scans easily exposes the patient to unhealthy amounts of radiation. The risk this poses is especially higher for children because their bodies are in development, which causes them to be more sensitive to changes from radiation. Receiving radiation in such an early stage of their lives mean they have a larger window to show the side effects than an adult. A study has reported that children who had cumulative radiation dose about 50 to 60 milligray (mGy) have shown a threefold increase in the risk of brain tumors [9]. Milligray (mGy) is another unit to estimate the absorbed dose of ionizing radiation. This amount of cumulative dose is gathered from $2-3$ head scans. The same study has shown that a similar dose to bone marrow increases the risk of leukemia by threefolds. 50 to 60 mGy of radiation is the result of 5 to 10 head CT scans [9]. Higher patient dosage is usually required to obtain a good spatial resolution, so reducing the dosage while maintaining good image quality is currently being researched.

The amount of radiation received by the patient can be altered by changing the parameters of the CT equipment such as tube current and voltage, pitch, slice thickness, and filters [10]. The tube current (mA) is linearly proportional to patient dose while tube voltage has an exponential relationship (kV) [10]. Decreasing either property will increase the amount of noise and lower image quality.

Systems in place to limit the amount of radiation include automated tube current modulation, which changes the current during CT scans [10]. It is programmed to scan each patient based on the appropriate amount of radiation required for the desired image quality in the region of interest [10]. Organs vary in thickness, so the current modulation accounts for the amount of x-ray attenuation and alters the current required during the CT scan.

The automated tube current system determines the current based on the specified noise level allowed and the "mAs per slice" which is calculated through Equation 2.3.

$$mAsPerSlice = mA * \frac{rotationtime}{pitch} \tag{2.3}$$

Pitch is defined as table travel per rotation/beam collimation [10]. As pitch increases patient dose decreases linearly [10].

There is a minimum slice thickness that is determined by the design of the detector

array [10]. Depending on the desired region, the slice thickness is changed to produce the optimum slices for images to reduce error [11]. Decreasing the slice thickness improves the z-axis resolution, but increases radiation exposure and scan time [10]. Changing the slice thickness affects the amount of noise since both have an inverse relationship [12]. Selecting an appropriate slice thickness ensures for good resolution and low noise levels. The slice thickness for clinical purposes typically ranges from 1 to 10 mm [12].

Bowtie filters are used to shape the x-ray beam to increase the intensity towards the center of the patient (region of interest) and reduce radiation on the peripheral [13]. The bowtie geometry allows for a more uniform distribution of x-ray beams, which improves image contrast and decreases scattering [13].

Other factors influencing image quality include misalignment of the patient from the isocenter of the CT gantry, which can cause artifacts and noise [10]. Larger patients require a greater amount of radiation to produce images due to larger attenuation. Radiologists aim to achieve good signal to noise ratio to produce viable images to be diagnosed.

2.4 Low Dose CT denoising Methods

As discussed in the previous section, there are some concerns about the radiation risks from CT scans. Using low dose computed tomography is one way to decrease radiation exposure. It can greatly reduce the cumulative radiation in patients who need regular screening, such as those in risk of lung cancer. While lung cancer can be fatal, if it is diagnosed in the early stages, the patient will have a higher chance of survival and living longer. However, the main problem in the path is that low dose CT images are very noisy and make it very difficult to detect abnormalities. To tackle this obstacle, many types of researches have been conducted in recent years.

Noise removal methods from low-dose CT scan images can be categorized into three groups: projection space denoising, iterative reconstruction, and reconstructed CT image denoising [14].

2.4.1 Projection Space Denoising

The projection data is a 2-D signal obtained from the CT scanner before reconstruction of CT image. Some studies have used this data (sinogram or raw data) to reduce the noise in low-dose CT images. Many of the methods in this group apply traditional image processing techniques on the sinogram data and then reconstruct the CT image by FBP or other methods. The noise statistics in the projection domain are well researched, and scientists have used this information to optimize the use of a non-linear filter to reduce the noise level. In a study, Manduca et al. have applied bilateral filtering on the projection data [15]. In this method, the intensity of each pixel is replaced by the weighted average of the neighborhood pixels, which are calculated based on their spatial proximity.

This method is capable of reducing the noise in the image while preserving the edges. However, experiments have shown that if the parameters are not optimized carefully, small edges can also be filtered out which leads to the loss of spatial resolution in the reconstructed CT image. System physics and photon statistics are incorporated in these methods that help to reduce the artifacts as well as the noise in the CT image; nevertheless, this makes the method depends on the CT scanner vendor. Another drawback is that these methods need the projection data which is not always available. This technique should be implemented on scanner reconstruction systems that tend to induce high cost.

2.4.2 Iterative Reconstruction

Iterative reconstruction methods have been used to reduce the noise of the low-dose CT images. In this group, the data is transferred multiple times between the projection and image domain then the algorithm tries to optimize the objective function. Figure 2.2 demonstrates how these techniques work. In the first step, the reconstructed CT image is used to generate projection data based on the system models and photon statistics. By comparing this data with the original data acquired from the CT scanner, it will be modified and then used to reconstruct a new CT image. This procedure is repeated multiple times until the objective function is optimized and then the final CT image will be produced.

These algorithms are divided into two main groups [14]: Full and Hybrid methods. The full methods such as Model-Based Iterative Reconstruction (MBIR) uses the detailed

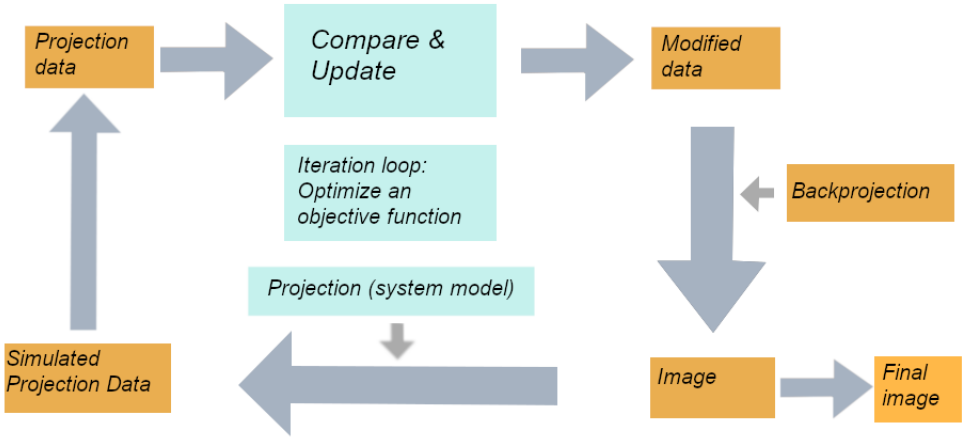


Figure 2.2: Iterative Reconstruction

model geometry, photon counting statistics, and x-ray beam spectrum [16]. Using all these aspects makes the procedure slow. In hybrid methods such as Sinogram Affirmed Iterative Reconstruction (SAFIRE), the noise reduction is performed in the image domain to projections, and the iterations between projection and image domain are only used to remove artifacts and improve the quality of the CT image [17, 18]. This group is faster compared to full reconstruction.

Since iterative reconstruction methods take into account system models and photon statistics, they reduce the noise better than the projection processing algorithms and also preserve the edges. Artifacts that are the result of photon starvation or metal plants can be removed from the final image. Although these iterations improve the performance, they greatly increase the computational cost. The other problems are these methods should be implemented on the scanner; they are vendor dependent, and they need the projection data. Another drawback is that iterative reconstructions change the noise texture compared to what radiologist are generally used to that may affect making diagnosis [14].

2.4.3 Reconstructed CT image Denoising

In this group, the noise removal tasks are done on the reconstructed CT image. Many of the techniques in this group are adapted from algorithms designed for natural image

denoising. Opposite to the previous methods, this group does not need the projection data and only processes the output image so they can be easily integrated with the workflow. They are fast and independent of the scanner vendor. Therefore, it is possible to use one denoising system for multiple scanners. By using post-processing methods to remove noise from CT images, it is a simpler process that does not require additional attachments to any CT equipment. It is more convenient to upload the images to this algorithm rather than replacing CT scanners and updating them. However, these methods do not incorporate system physics and do not remove artifacts very well. Some of the proposed methods in this domain are explained below.

Sparse representation

In sparse representation, the goal is to construct an input data as a combination of a few atoms from an overcomplete dictionary. The optimization problem is solved in Equation 2.4.

$$< D, X >= arg_{D,X} min||Y - DX||_2^2, s.t. \forall i ||x_i|| \leq T \tag{2.4}$$

Here, $Y = [y_1, y_2, ..., y_N]$ are N source signals. To represent these signals by sparse coding we should find the sparse representation $X = [x_1, x_2, ..., x_N]$ and also learn the overcomplete dictionary $D = [d_1, d_2, ..., d_K]$ with K columns that minimize the reconstruction error $||Y - DX||_2^2$. T is the sparsity constraint and forces the number of non-zero entries in each x_i to be not greater than T. This also means to construct each y_i; we can use not more than T atoms from the dictionary.

The first step in solving this optimization problem is to find the sparse code X of Y with respect to a fixed dictionary D. For this purpose, matching pursuit [19], orthogonal matching pursuit [20] or similar algorithms are used.

In the next step the dictionary D should be updated. Different algorithms are developed to perform this such as method of Method of Optimal Direction (MOD) [21] or K-Singular Value Decomposition (K-SVD) [22]. To learn the dictionary by MOD method, we use $D = YX_k^+$ where X_k^+ is pseudo-inverse of X at kth iteration. The computational cost of calculating this inverse can be very high since most of the time X_k is a large matrix. K-SVD is another method that offers an efficient algorithm to learn the dictionary D. In this method, each column of the dictionary gets updated one at a time

by finding the rank-1 approximation with singular value decomposition.

Sparse representation has been widely used in different image processing fields such as image denoising, image classification, and image restoration with successful results. It also has been adopted to remove noise from low-dose CT images. A fast dictionary learning method has been developed to improve abdomen tumor low-dose CT images[23]. Abhari et al. has proposed an advanced K-SVD to enhance low-dose CT images [24].

BM3D

Block Matching and 3D Filtering (BM3D) is an algorithm proposed to remove noise from natural images. In this technique, patches of an image are compared to a reference patch from the image [2]. The patches that are below the reference threshold of dissimilarity are grouped and form a 3D cylindrical shape. Then a 3D transform is used to convert the 3D group to a wavelet transform. BM3D is usually performed in 2 steps. In the first step, hard thresholding is applied to the result of the 3D transformation then an inverse transform and aggregation are performed to acquire a basic estimation. In the second step, this basic estimation is used to find similar patches to the reference fragment and creates another 3D array. The new array is combined with the 3D array from the previous step, and step 1 is repeated. The only difference during this iteration in comparison to step 1 is that the Wiener filter is used instead of hard thresholding. Figure 2.3 shows this procedure.

Bm3D is considered the state of the art denoising algorithm that proved to perform well even when the image is very noisy. It has been used for deblurring and image restoration. Based on this idea, some studies have developed algorithms to remove noise from the low-dose CT images with successful results [25, 26, 27].

Deep learning

Deep learning is a part of machine learning family that has become very popular in recent years. Although the basics of deep learning were developed many years ago, the high computational capacity of new GPUs in combination with techniques like residual learning [28] and batch normalization [29] have made training of deep networks possible. By using deep neural networks, scientists have achieved outstanding results in tasks such as image processing and Natural Language Processing (NLP). In image processing, some

of the proposed convolutional neural networks have outperformed traditional methods in image recognition, semantic segmentation, and image restoration. Deep learning has been proved effective when applied to medical images. Chen et al. have used a 3-layer convolutional neural network for low-dose CT denoising [30]. This network was initially proposed to remove noise from natural images [31]. According to the authors, the network performs sparse coding and outperforms K-SVD and BM3D [30]. Later, the authors developed a convolutional auto-encoder network [32] with residual learning that provided better performance compared to their previous work. Nishio et al. has conducted one of the first studies in the field and uses auto-encoders to restore low-dose images [33]. The input and output of these networks are low-dose and normal-dose CT images, respectively. However, Kang et al. has transformed the images to the wavelet domain and then applied the coefficients to his proposed network [34].

In 2014, Goodfellow introduced Generative Adversarial Networks (GAN) [35] and it has been prevalent among researchers. The network consists of two main part, a generative network (G) and a discriminator (D) that can be seen in Figure 2.4. The generative G tries to construct images similar to the target from random data. The discriminator is responsible for distinguishing between real and fake data from the generator. The main

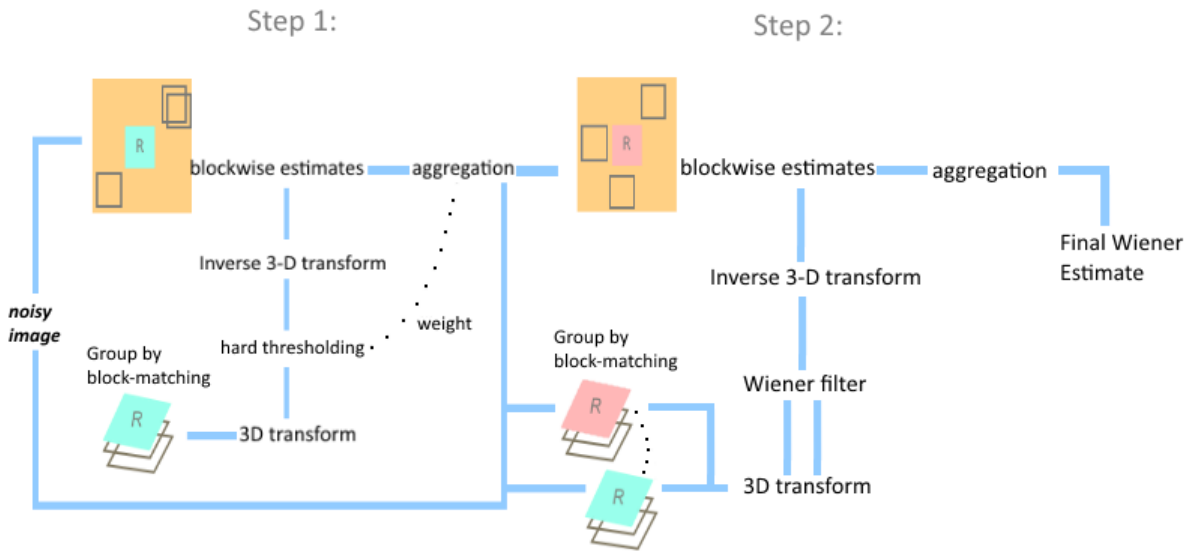


Figure 2.3: Block Matching 3D

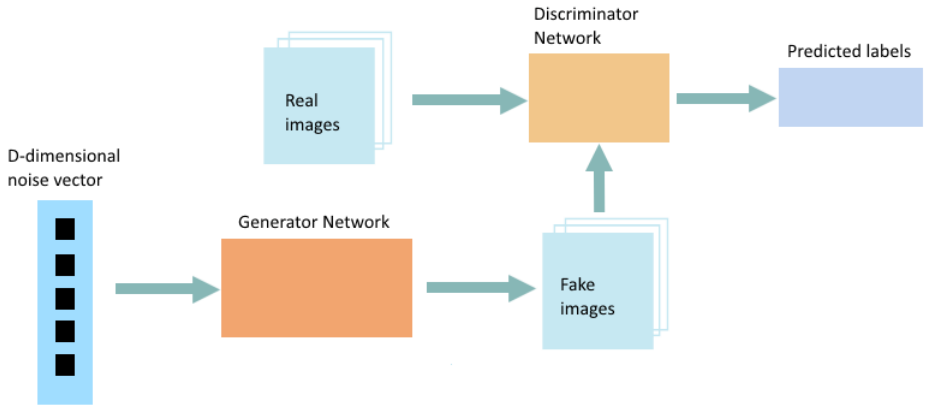


Figure 2.4: Generative Adversarial Networks

problem is to solve a min-max optimization where the generator tries to minimize the objective function while the discriminator maximizes it. When the network is trained well, the generator will be able to pass off fake data, which the discriminator will recognize as real data. Networks from this group have been used to tackle different tasks such as text-to-image synthesis, image-to-image translation, face editing, prediction of future video frames, and image super-resolution.

In some studies, a low-dose CT image is given to the generator to produce an image with a better quality that the discriminator classifies it as a normal-dose CT image [36, 37, 38]. In most of the convolutional networks designed for low-dose CT noise removal, the objective function is the mean square error between the output of the network and the ground truth. However, in generative adversarial networks, the perceptual loss in combination with Wasserstein distance is used to optimizes the GAN. Yi et al. have proposed a generative adversarial network in conjunction with a third network that enables sharpness detection [36]. The sharpness detection network produces a sharpness map of the generator output and ground truth, then the mean square error between these two are added to the objective function.

Overall, researchers have shown that many algorithms employing neural networks outperform traditional methods in denoising low-dose CT images. Deep learning had made it possible to achieve higher quantitative and qualitative results in medical imaging.

The proposed method in this study performs noise reduction over the reconstructed

CT image and uses deep neural networks to achieve this goal.

Chapter 3

Deep Learning

3.1 Basics of Neural Networks

Neural networks have been around for almost 60 years, but recently scientists have discovered its greater potential. Technological advancements such as facial recognition and voice command are made possible due to neural networks. Enhancements to data processing and computation in conjunction with the massive data collection from social media has allowed neural networks to be extensively developed. In general, a neural network consists of multiple layers with some nodes (neurons) in each layer. Figure 3.1 shows a typical structure of a simple neural network and connections between layers.

The neural network is a field of machine learning that learns a function to generate

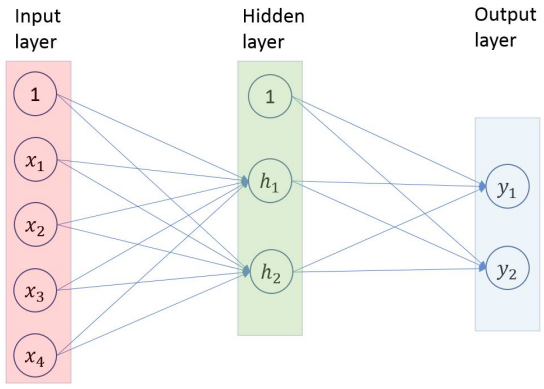


Figure 3.1: Neural Network with 1 hidden layer

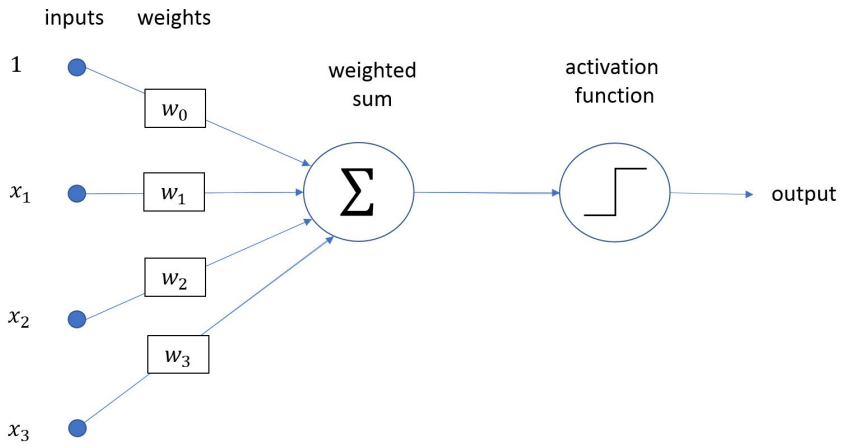


Figure 3.2: Inputs and output of a neuron

the desired output based on the input. Similar to other machine learning techniques, a neural network learns from training data and label pairs, and tries to predict labels for unseen data. In neural networks, the training data is applied to the neurons in the first layer; then the result is produced in the output layer. The output result from the last layer is tested to determine its accuracy against the expected results. Figure 3.2 shows a neuron model. For each layer the output of a neuron is defined as function of linear systems.

$$output = f\left(\sum_{i=1}^{N}(w_i x_i + b)\right) \tag{3.1}$$

In Equation 3.1, $x = (x_1, x_2, ..., x_N)$, $w = (w_0, w_1, w_2, ..., w_N)$ and N represent data , weights and the dimension of x, respectively. f is an activation function such as Sigmoid, tanh or ReLU. This equation calculates the output value for a neuron when the input x with N dimension is applied to it. x_i in equation 3.1 should not be confused with (x_i, y_i) that represent a pair of training data and label and is the general notation in machine learning. In the rest of this document, also, (x_i, y_i) is used to show a training sample.

The objective of training a neural network is to find values for the weights that allow the network to produce the label (y_i) when the training sample (x_i) is applied to the system. By minimizing the objective function J, the network can more accurately produce the desired results. Different objective functions exist for different applications.

Cross-entropy is a common choice in classification applications, whereas the mean square error is used for image super-resolution, denoising, and inpainting. Cross-entropy and mean square error can be calculated with Equations 3.2 and 3.3, respectively.

$$J(Y, \hat{Y}; W) = -\frac{1}{N} \sum_{i=1}^{N} y_i \log \hat{y}_i \tag{3.2}$$

$$J(Y, \hat{Y}; W) = ||Y - \hat{Y}||_2^2 = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} (y_i - \hat{y}_i)_2^2 \tag{3.3}$$

In both equations, y and $\hat{y}$ are the label and the output of the network respectively. N is the number of samples, and W is the set of all the weights in the network. To optimize the objective function and find the weights, we use gradient descent algorithm or improved version of it.

3.2 Optimization Algorithms

3.2.1 Gradient Descent

Gradient Descent is an iterative optimization algorithm that minimizes the objective function, $J(W)$, by updating parameters. The algorithm is:

1. Initialize w_j

2. Repeat until convergence {

$$w_j = w_j - \alpha \frac{\partial J(Y, \hat{Y}, W)}{\partial (w_j)}$$

}

Here, w_j is the weight that we want to update and α is the learning rate. If the objective function is mean square error, the gradient descent equation will be as follows:

$$w_j = w_j - \alpha \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i) x_i \tag{3.4}$$

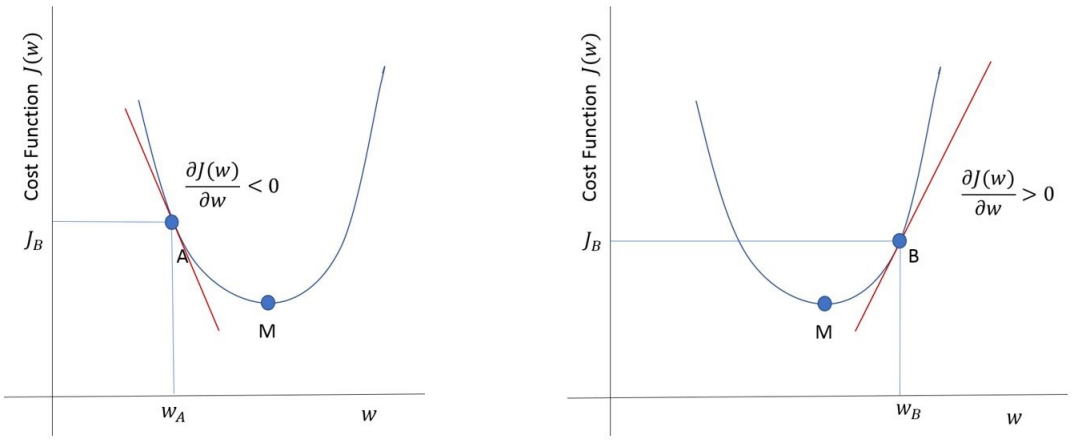


Figure 3.3: Updating w in Gradient Descent

Figure 3.3 demonstrates how gradient descent works when we have just one weight. For a convex function, if the current weight (w) is less than the optimum value then the gradient ($\frac{\partial(J)}{\partial(w)}$) will be negative and since the learning rate (α) is positive, the weight (w) will be increased. On the other hand, if the weight is higher than the optimum weight, then the gradient will be positive, and the weight will be decreased. This algorithm moves the weight in the correct direction by using the tangent to find the direction towards the minima of the equation. The weights are adjusted to the minimum of the function.

Learning rate is a hyperparameter that should be chosen carefully. If the learning rate is too low, it will take a long time to find the point (slow convergence), and it is possible to get stuck in local minima. On the other hand, a high learning rate may lead to divergence and prevent the gradient descent from finding the optimum weights.

3.2.2 Stochastic Gradient Descent

In a neural network such as other machine learning methods, often there will be extensive amounts of data in the training set. By looking at equation 3.4, it is clear that the computation will be affected if the matrix containing the training data is too large and requires substantial memory storage. Therefore, calculating and minimizing the cost for all training samples is generally avoided. Stochastic Gradient Descent (SGD) solves this problem by calculating the cost function for one sample at a time. The gradient of the cost function is used to update the weights. Equation 3.5 shows this change.

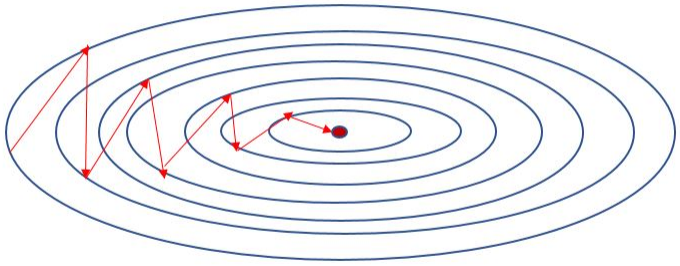


Figure 3.4: Searching for the optimum by gradient descent algorithm

$$w_j = w_j - \alpha(y_i - \hat{y}_i)x_i \tag{3.5}$$

Although SGD has low computation cost and uses a small amount of memory, the achieved results are not very accurate. Mini-batch gradient descent is a compromise between the above algorithms. It has a higher accuracy rate than SGD and requires less memory. In this method, cost calculation and updating the weights are done over a mini-batch of the training set. Usually, there are 32 to 128 examples in a batch. In most papers and programming languages, this method is referred to as SGD. Opposite to one might think, increasing the batch size does not always improve the performance. It has shown that using large batches lead to degradation [39]. It may be the result of landing on a sharp minimum rather than the optimum one.

3.2.3 Gradient Descent with Momentum

The problem with gradient descent is that it applies the same learning rate in all direction. By investigating figure3.4, it can be seen that a higher learning rate is needed in the horizontal direction and smaller learning rate in the vertical direction. If we choose a large learning rate, overshooting may occur in the vertical dimension which will prevent the network from convergence. If a small learning rate is chosen to avoid the mentioned problem, then the learning process will be prolonged.

To solve this problem, gradient descent with momentum is proposed in [40], which accelerates learning in the direction with more changes.

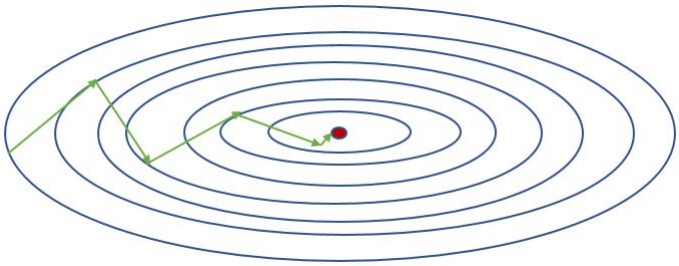


Figure 3.5: Gradient descent with momentum

$$v_t = \beta v_{t-1} + (1-\beta)\nabla_{w_j} J(w_j)$$
$$w_j = w_j - \alpha v_t$$

β represents the momentum and it adds a fraction of the past value of v_{t-1} to the current update. As the equation shows, this method finds the exponentially weighted average of the previous gradient values. As Figure 3.4 demonstrates, the gradient in the vertical direction is constantly positive and negative, which makes the average close to zero. Therefore, the learning in this direction becomes slower. On the other hand, in the horizontal direction, the moving average leads to a higher learning rate and accelerates learning. The typical value for β is 0.9. Figure 3.4 shows how learning is changed when using momentum gradient descent.

3.2.4 Adam

Adaptive Momentum Estimation (Adam) is a very powerful optimizer that has performed well in many deep networks [41].It takes advantages of the momentum as used in momentum gradient descent in addition to the exponentially weighted average of past squared gradients v_t.

$$
\begin{aligned}
m_t &= \beta_1 m_{t-1} + (1-\beta_1)\nabla_{w_j} J(w_j) \\
v_t &= \beta_2 v_{t-1} + (1-\beta_2)\nabla_{w_j} J^2(w_j) \\
\hat{m}_t &= m_t/(1-\beta_1^t) \\
\hat{v}_t &= v_t/(1-\beta_2^t) \\
w_j &= w_j - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}}
\end{aligned}
$$

In this equation, m_t and v_t represent mean and uncentered variance of the gradient and β_1 and β_2 are the decay rates for these moments. The recommended values for $beta_1$,$beta_2$, and ϵ are 0.9, 0.999, and 10^{-8}, respectively. It should be noted that α is to be found with tuning.

Adam optimization algorithm is easy to implement, computationally efficient and requires little memory. It is also not very sensitive to hyper-parameters and usually converges faster compared to the gradient descent as higher learning rate can be chosen. Adam has become very popular and now is used widely to train deep network.

3.3 Convolutional Neural Network

Convolutional Neural Network (CNN) shares many concepts with a neural network. It is mostly used for images because the number of weights can grow drastically when pixels of an image are used as information (called features) of the training data. This makes neural network impractical in the field of image processing. Convolutional neural networks have shown great results in image recognition, classification, and super-resolution. There are four main operations in a convolutional neural network as shown in Figure 3.6 [42].

1. Convolution
2. Non Linearity (ReLU)
3. Pooling or Sub-Sampling
4. Classification (Fully Connected Layer)

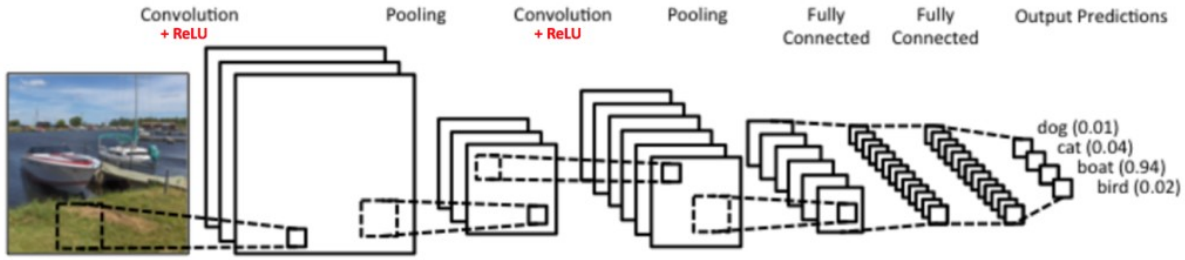


Figure 3.6: Convolutional neural network [42]

3.3.1 Convolutional operation

Convolution operation is used to capture the local dependencies in an image. Figure 3.7 demonstrates this operation. Every image can be written as a matrix of pixel values. Here, the convolution of a 5×5 image with pixel values 0 and 1 and a filter size 3×3 is calculated. First, element-wise multiplication between the two matrices is computed, and then the multiplication outputs are added together to get the final result. This number is the value of the first element of the output matrix. In other words, this calculation provides a number for a 3×3 patch of the image. Next, the filter is moved vertically and horizontally to the next pixel and calculates the other output elements similarly.

It is possible to slide the filter by more than one pixel (stride>1) which will result in a smaller output matrix. The output matrix is called a feature map or convolved feature. Different filters will provide different feature maps, and each of them extracts a special feature from the input image such as vertical lines or special curves of the picture. In practice, elements of each filter will be learned during the training process. However, some parameters of the network such as the number of filters in each layer, the size of

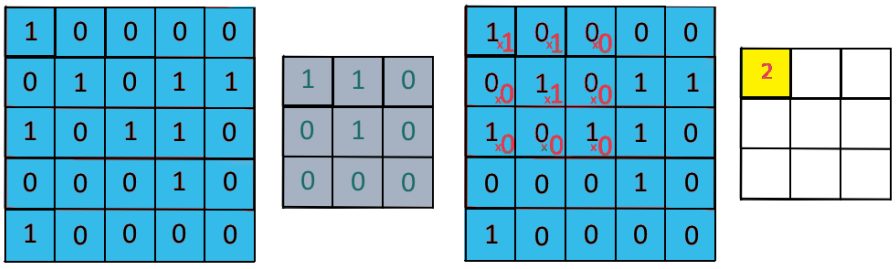


Figure 3.7: Convolutional operation

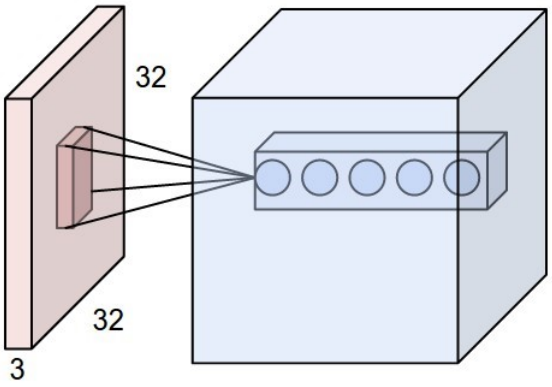


Figure 3.8: Applying 5 filters to an RGB image results in a feature map with depth equal to 5 [42]

the filters, stride, and architecture of the network should be precisely defined beforehand. As mentioned before, it is possible to apply more than one filter to the image and create multiple feature maps. The number of filters is equal to the depth of the output. Figure 3.8 shows this concept.

3.3.2 Rectified Linear Unit

Rectified Linear Unit (ReLU) is the most common activation function in convolutional networks [42], and it is applied to the result of the convolution operation. It zeroes out all negative values in a feature map such that the feature does not contribute to the algorithm. It adds to the sparsity of the feature map and helps with optimization because it prevents the algorithm from being trapped in local minimums. Figure 3.9 shows this function. Generally, the combination of convolution and ReLU is considered one layer and a network can have multiple of these layers in succession.

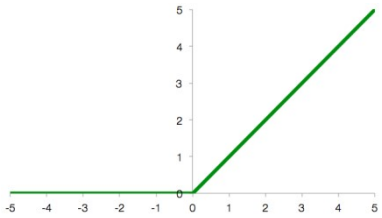


Figure 3.9: Rectified Linear Unit

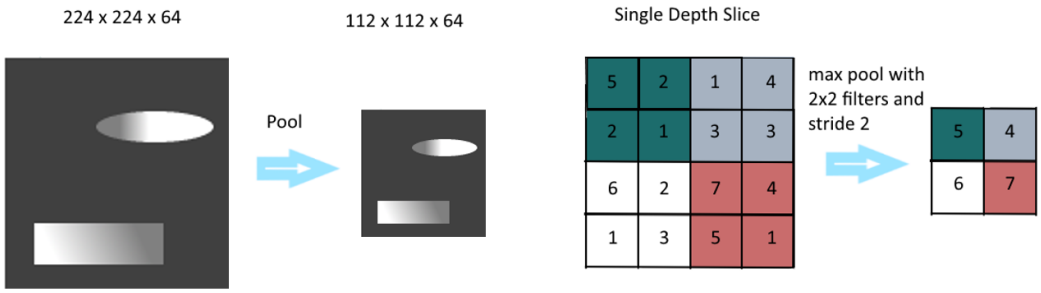


Figure 3.10: Pooling layer

3.3.3 Pooling

Pooling or sub-sampling reduces the dimension of the feature map but keeps the most important information. This operation replaces a block (for example 2×2) by the average of its elements or the maximum value. Figure 3.10 demonstrates the dimension reduction using the maximum value. Pooling helps to progressively reduce the spatial size of the feature map and reduces the number of parameters and computation in the network.

3.3.4 Fully Connected Layer

This layer is similar to a regular layer in the neural network. Every neuron in this layer is connected to all the neurons of the next layer.

3.3.5 Backpropagation

Backpropagation is an algorithm to train neural networks and is used in conjunction with an optimization method such as gradient descent. It helps us to understand how changing each weight affects the objective function. To update the weights of a convolutional neural network with multiple layers we follow below steps [43]:

1. Initialize weights and bias.

2. First input (x_1) is applied to the first layer.

3. Feed forward: for each layer $l = 2, 3, \ldots, L$ compute Equation 3.6

$$Z_x^l = W^l * \ f\left(Z_x^{l-1}\right) + b^l \tag{3.6}$$

Where, Z_x^l is the output of each layer to input x, f is the activation function (ReLU), and W^l, b^l are filter weights and bias term in each layer. This step calculates the output of each layer in the network to our input.

4. Calculate $f'\left(Z_x^l\right)$ for each layer.

5. Compute the output error. J is the objective function.

$$\delta^L = \frac{\partial J(x^L, y)}{\partial x^L}. \ f'(Z_x^L) \tag{3.7}$$

6. Backpropagate the error: For each $l = L-1, L-2, \ldots, 2$ compute

$$\delta_x^l = \ \delta_x^{l+1} * ROT180\left(W_x^{l+1}\right) f'(Z_x^L) \tag{3.8}$$

Here, ROT means 180 degree rotation. At this step, the error is propagated back to the network. δ_x^l Shows the errors at each layer with the current weights.

7. The gradient of the cost function can be calculated by Equation 3.9

$$\frac{\partial J}{\partial w_j} = \delta_x^l * f(ROT180(Z_x^{l-1})) \tag{3.9}$$

This term is used to update the weights in stochastic gradient descent.

3.4 Image Super-Resolution with CNN

Image super-resolution is a classical problem in image processing. The goal is to construct a high-resolution image using a low-resolution image. Here, we study research that has proposed a simple three-layer convolutional network to solve this problem [31]. This

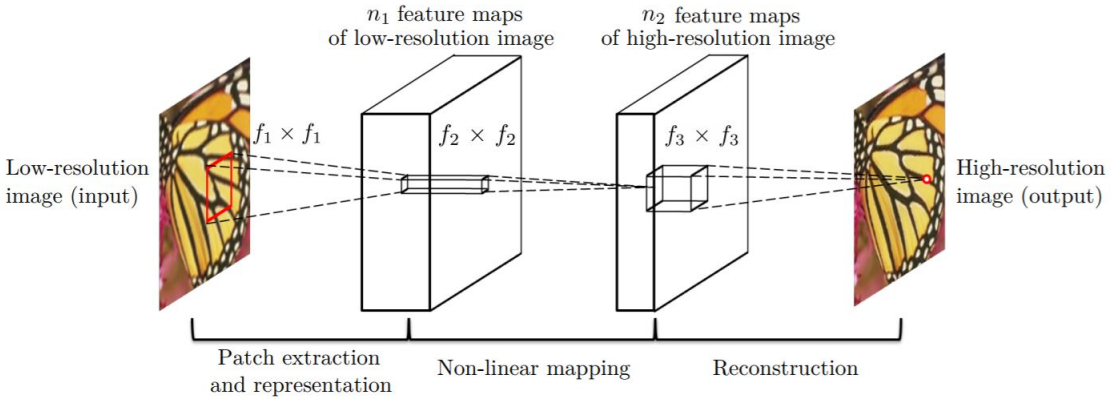


Figure 3.11: Architecture of 3 layer CNN for image super-resolution [31]

network takes a low-resolution image as an input and reconstructs a clear image in the output. It offers an end-to-end solution by using a convolutional neural network to eliminate noise. The network consists of three convolutional layers as shown in Figure 3.11. Authors explain that these three layers perform patch extraction, non-linear mapping, and reconstruction, subsequently.

3.4.1 Patch Encoding

The first layer of the proposed network consists of a convolution operation and ReLU.

$$C_1(y) = ReLU(W_1 * y + b_1) \tag{3.10}$$

In equation 3.10, W_1, b_1 and $C_1(y)$ represent filters, biases and output of the first layer, respectively. W_1 consists of n_1 filter of size $s_1 \times s_1$. This operation extracts $s_1 \times s_1$ patches of the image and applies n_1 filters on it then adds a bias term. The output has n_1 feature maps. ReLU is applied to these outputs to obtain the results. Figure 3.12 shows what happens to each patch in this layer.

Sparse Representation is a popular method for image restoration. In this method, patches are extracted from the original image, and they are then represented by dictionaries. It is similar to transforming the information from the image domain to another domain. These dictionaries can be formed by a predefined format like Discrete Cosine Transform (DCT) and wavelet, or they can be calculated and optimized during the program (K-SVD). The authors of this paper exclaim that the convolutional operation in the

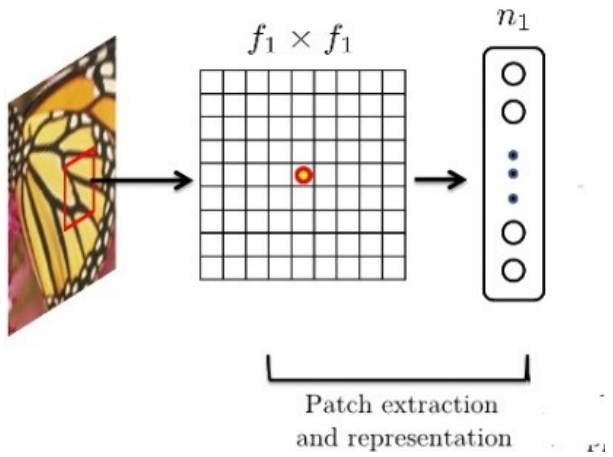


Figure 3.12: Result of applying n_1 filter to each patch [31]

first layer is similar to sparse representation. They consider filters as dictionary atoms and explain that projecting a patch of the image onto a dictionary is equal to applying filters on that image.

3.4.2 Non-Linear Filtering

In the second layer n_2 filters of size $s_2 \times s_2$ are applied to the output of the first layer. These filters are shown by W_2 in the following equation:

$$C_2(y) = ReLU(W_2 * C_1(y) + b_2) \tag{3.11}$$

The authors believe that the output of this layer can be considered as denoised patches in a sparse representation. This paper uses only two layers of the convolutional neural network to improve the quality of the picture. It is possible to use more layers which probably could improve the performance, but it adds to the complexity of the network.

3.4.3 Reconstruction

This is another convolutional layer but only uses one filter of size $s_3 \times s_3$.

$$C_3(y) = (W_3 * C_3(y) + b_3) \tag{3.12}$$

The output of this layer is a full-size denoised image. The authors argue that this step is similar to merging patches in other methods. It takes the weighted average between overlapping patches and constructs a full image.

Sizes and number of filters in each layer are as follows:

- W_1: n_1 filter with size $s_1 \times s_1$, $s_1 = 9$, $n_1 = 64$
- W_2: n_2 filter with size $s_2 \times s_2$, $s_2 = 3$, $n_2 = 32$
- W_3: 1 filter with size $s_3 \times s_3$, $s_3 = 5$

Authors of this paper compare the results of the proposed network with BM3D and KSVD. They exclaim that the network outperforms other methods.

3.5 Autoencoders

Autoencoders are a group of neural networks that the output of the network is equal to the input. A basic autoencoder consists of three layers: input, output, and a hidden layer. Usually, the number of neurons in the hidden layer is less than the neurons in the input and output layers. Figure 3.13 shows such an encoder. The objective function that should be optimized is:

$$J(X, \hat{X}) = ||X - \hat{X}||_2^2 = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} (x_i - \hat{x_i})_2^2 \tag{3.13}$$

where, X and $\hat{X}$ are input and output. In Figure 3.13 the input and output layer have four nodes while the hidden layer has only two neurons. This means that the network should learn a representation of the input with fewer features. If there is a correlation between the features, then the autoencoder can extract this low dimensional representation. The output of the network is reconstructed from these two features and should be similar to the input. Therefore, a simple autoencoder works like a Principal Component Analysis (PCA). In fact, if the network does not have any non-linearity (caused by activation function), then the results obtained from the hidden layer in an autoencoder network are very similar to PCA. Mapping from X to $h(X)$ and from $h(X)$ to $\hat{X}$ in the autoencoder is as follows,

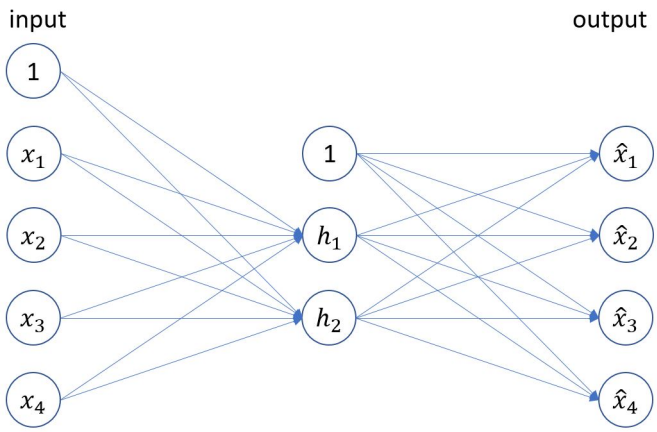


Figure 3.13: 3 layer autoencoder

$$h(X) = WX, \quad \hat{X} = W'h(X) \tag{3.14}$$

and in PCA, the first k principle components are found from Equation 3.15,

$$h(X) = (\Sigma_k^{-1} U_k^T)X \tag{3.15}$$

by comparing Equation 3.13 to 3.15 the weights can be found:

$$W = \Sigma_k^{-1} U_k^T, \quad W' = U_k \Sigma_k$$

3.5.1 Denoising Autoencoder

In practice, to prevent the network from learning just an identity function, Denoising Autoencoders (DA) are used. For this purpose, a corrupted version of the input is feed to the network, and the autoencoder is forced to provide a clean image in the output. Corrupting the input x is done by adding noise to the original image y or by replacing some pixels with zero.

$$x = \eta(y)$$

Figure 3.14 displays the architecture of DA. The results show that such a network

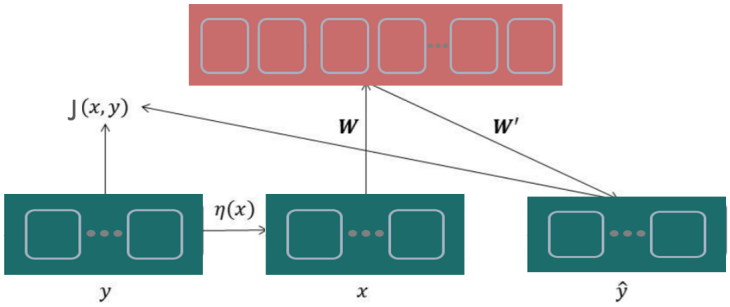


Figure 3.14: Denoising autoencoder architecture

can extract more robust features in the hidden layer [44]. For example, experiments demonstrate that if handwritten digits are directly fed to an autoencoder, the output of the hidden layer does not provide that much information, but adding noise to the input will result in more meaningful results in the hidden layer output, including strokes and arcs that make handwritten digits. Denoising autoencoders are also used for image super resolution or noise removal problems. If input x is the noisy image and y is the ground-truth, then the following equations can be written:

$$h(x_i) = \sigma(Wx_i + b) \tag{3.16}$$

$$\hat{y}(x_i) = \sigma(W'h(x_i) + b') \tag{3.17}$$

where, σ is a Sigmoid function. W, b, W' and b' are the weights and biases for the first and second layers, respectively. The network should learn $\Theta = \{W, b, W', b'\}$ that satisfies the minimization of the difference between output and target.

$$\theta = arg_\theta min \sum_{i=1}^{N} ||y_i - \hat{y}_i||_2^2$$

3.5.2 Sparse Denoising Autoencoders

Not all the autoencoders have fewer nodes in the hidden layer. Sparse Denoising Autoencoders have more neurons in the hidden layer than input and output layers. This group can also provide interesting information about the structure of the input. This is

achieved by adding the sparsity constraint to the hidden layer [45]. The sparsity constraint means that some of the neurons in the hidden layer should be inactive most of the time. If the activation function is Sigmoid, the output of the neuron will be close to 1 for being active and almost 0 if the neuron is inactive. If $h_j(x_i)$ is the result of activation of node j in the hidden layer when input x_i is given to the network, the average activation of node j over all the inputs will be calculated using Equation 3.18.

$$\hat{\rho}_j = \frac{1}{N}\sum_{i=1}^{N} h_j(x_i) \tag{3.18}$$

The sparsity constraint enforces $\hat{\rho}_j$ to be equal to a predefined sparsity parameter ρ using Kullback-Leibler (KL) divergence criterion. KL measures the difference between two probability distribution.

$$KL(\hat{\rho}||\rho) = \sum_{j=1}^{N} \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho)\log \frac{1-\rho}{1-\hat{\rho}_j}$$

If $\hat{\rho}_j$ is equal or close to ρ, then the KL will be almost zero. The new objective function J_{sparse} for a sparse autoencoder is defined as follows,

$$J_{sparse}(X, Y; \theta) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{2}||x_i - y_i||_2^2 + \beta KL(\hat{\rho}||\rho) + \frac{\lambda}{2}(||W||_F^2 + ||W'||_F^2)$$

The objective function is optimized by back-propagation. If ρ is chosen to be small, many of the nodes in the hidden layer will be zero.

3.5.3 Stacked Autoencoders

All the networks discussed so far have one hidden layer. Vincent et al. [46] have proposed a deep network by stacking more layers to improve the performance. In this network, first of all, a three-layer DA is trained as usual. Through this process weights W_1, b_1, W_1' and b_1' will be learned. In the second step, the output of the hidden layer from the previous step ($h(x_i) = \sigma(W_1 x_i + b_1)$) is collected and used as the input for the next DA, where

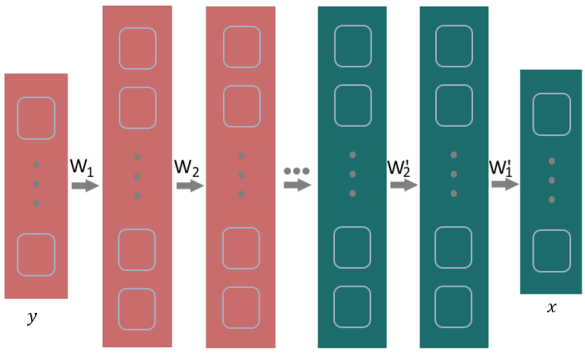


Figure 3.15: Denoising autoencoder architecture

the target is $h(y_i) = \sigma(W_1 y_i + b_1)$. By using this data, a new DA is trained. The result of this training is W_2, b_2, W_2' and b_2' . More layers can be added by repeating the second step.

After finding all the weights, the stacked denoising autoencoder is initialized with these weights. Figure 3.15 shows the architecture of the stacked denoising autoencoder. The next step is fine-tuning to find the optimum parameters when all the layers are in the network. Here, the objective function is

$$J_{sparse}(X, Y; \theta) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} ||x_i - y_i||_2^2 + \frac{\lambda}{2}(||W||_F^2 + ||W'||_F^2)$$

3.6 Batch Normalization

Researchers believe that adding more layers to a neural network is a key to achieve better performance. However, deep networks are generally hard to train. One problem is that the input distribution of each layer changes in each training step. This is mainly because in the training process the parameters of the previous layers change. This problem makes the gradients in the backpropagation step either very small or very large and to avoid that, hyper-parameters should be chosen very carefully. Since the vanishing/exploding

gradient prevents the network from convergence, the learning rate should be small that makes the training very slow. Ioffe et al. [29] called this phenomenon covariant shift and proposed Batch Normalization (BN) to solve this problem.

In batch normalization, the output of each layer is normalized (zero mean, unit standard deviation), similar to what is usually done for the inputs of the first layer. This transformation is performed over each mini-batch, which is m inputs x_i of an activation function. We can find the batch normalization transform result y_i as follows,

$$\mu_\beta = \frac{1}{m}\sum_{i=1}^{m} x_i$$
$$\sigma_\beta^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_\beta)^2$$
$$\hat{x_i} = \frac{x_i - \mu_\beta}{\sqrt{\sigma_\beta^2 + \epsilon}}$$
$$y_i = \lambda \hat{x_i} + \beta$$

here, m is the number of samples in a mini-batch. μ_β and σ_β^2 are the mean and variance of the mini-batch. ϵ is a constant added to the mini-batch variance for numerical stability, and parameters β and λ are to be learned. As it comes from the name of the method, this normalization is done over each batch. In the mentioned study, the normalization is done after applying the convolution filter and before the activation function. However, some researchers have used BN after activation function and just before feeding to the next layer, too. BN increases accuracy while reducing the training time. With this method, it is possible to use higher learning rates to increase training speed.

3.7 Residual learning

It has been found that when a network gets very deep, for example over 18 layers, the accuracy decreases even after using BN and solving convergence problems. This issue is not the result of overfitting because researchers have found that in a deep network even the training error grows. Deep residual learning for image recognition proposes an algorithm that can solve degradation issue [28]. Figure 3.16 shows the building block of

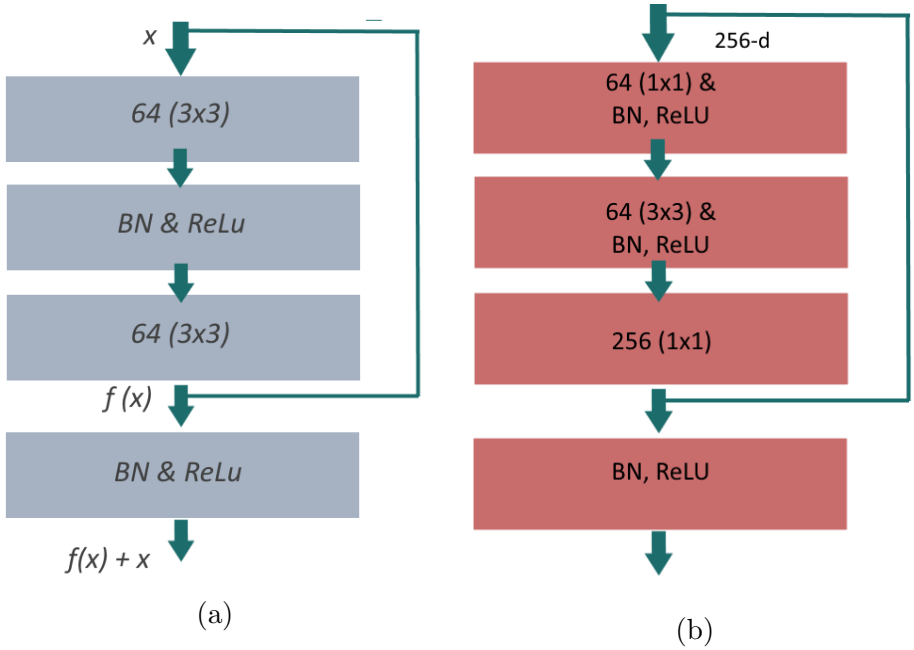


Figure 3.16: Residual network building block (a) two-layer building block, (b) trhee-layer (bottleneck) building block

this method.

The fundamental idea of this algorithm is to feed-forward input (X) and add it to the output of two consecutive layers ($f(X)$). Therefore the outcome of this block will be

$$Y = f(X) + X$$
$$Y = W_2\sigma(W_1X) + X$$

here, W_1 and W_2 are the weights of the first and second layers. Since the inner part of this convolutional network predicts the difference between the target and input, authors call their algorithm residual learning. They argue that optimizing this network is simpler than a similar network without the shortcut. For example, if the goal is to find $Y = X$, it is much simpler to force the residual to be zero than learn identity mapping through non-linear convolutional layers. It is worth noting that this shortcut improves performance without adding extra parameters or computational cost. An important point to keep in mind is that $f(X)$ and X should have equal dimensions. If the sizes do not match, a

projection of X should be made by multiplying it to a matrix.

To build a deep network, He. et al. [28] stack multiples of these blocks and show that the accuracy is higher than when no shortcut is used. Figure 3.17 shows the network that is used for classification purpose and uses 3.16a block.

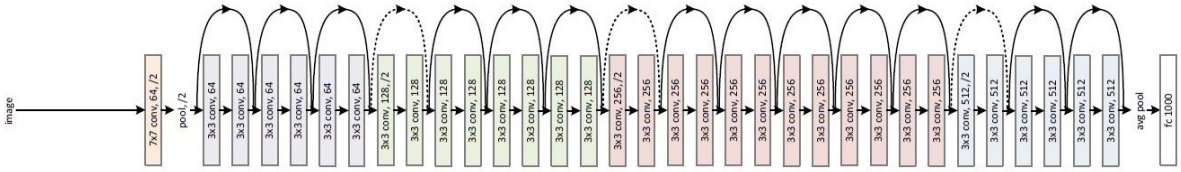


Figure 3.17: 34 layer residual network [28]

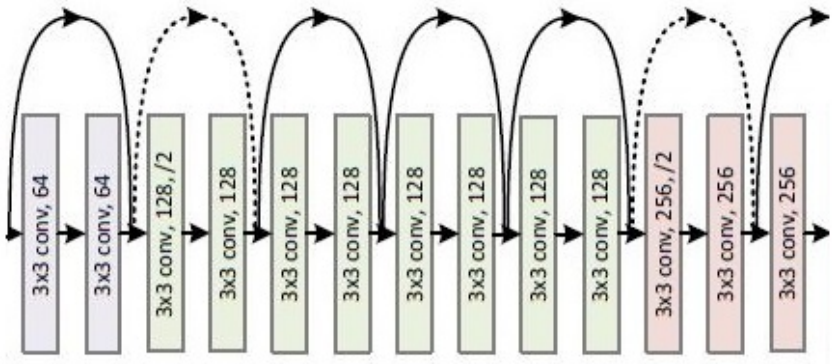


Figure 3.18: Enlarged section of 34 layer residual network [28]

Implementing very deep networks with the two-layer block of Figure 3.16a is very time-consuming. To tackle this difficulty, authors use a three-layer bottleneck block (Figure3.16b). In this new block, the first layer reduces the number of feature maps from 256 to 64. Then 3×3 convolution layer is applied to this smaller size input. The final layer restores the dimension to 256. Authors were able to develop networks with 150 layers with this bottleneck block and also achieve higher gain and convergence rate.

Many researchers have employed neural networks and developed different architectures for a variety of applications. K. Zhang et a. [47] use both batch normalization and residual learning and propose a network for noise removal. They discuss that their algorithm outperforms other denoising methods such as BM3D [2], WNNM [48], and EPLL [49]. Figure 3.19 shows this network. The output of this network is $R(Y) = Y - X$

where X and Y are clean and noisy images, respectively, and $R(Y)$ is the noise. The cost function is

$$J(\theta) = \frac{1}{2N} \sum_{i=1}^{N} ||r(y_i; \theta) - (y_i - x_i)||_F^2 \tag{3.19}$$

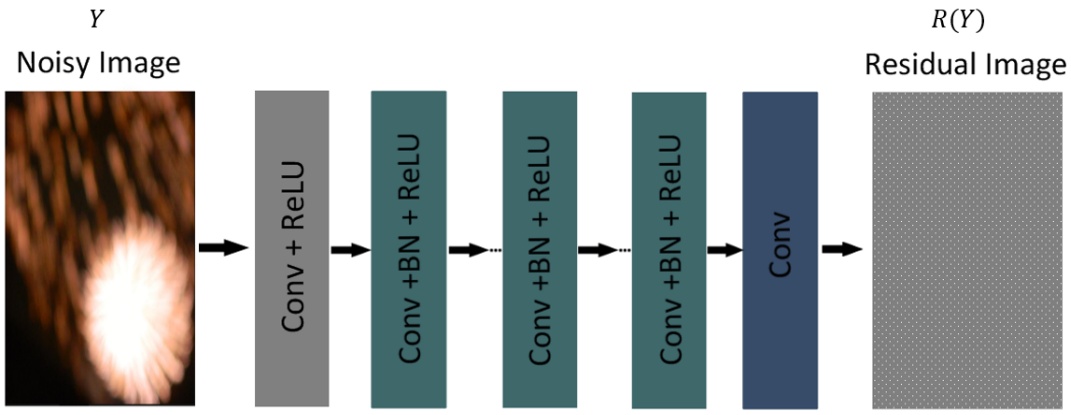


Figure 3.19: Denoising CNN architecture [47]

All the filters of this network are (3×3), and each layer produces 64 feature maps. Batch normalization is done in the middle layers after convolution and before activation. They have proposed two networks with 17 and 20 layers to remove noise with a specific level and unknown level, respectively. To train the network, 40×40 overlapping patches from the images are extracted. They have shown that the network is capable of removing noise from images even when the noise level is unknown.

Another network that is proposed for image restoration combines the idea of residual learning and autoencoders [50]. This paper develops a three deep CNN with 10, 20, and 30 layers. The first half of the layers in these networks are convolutional layers, which extracts primary components of an image and removes noise. The second half is deconvolutional layers, also called transpose convolutional, that merges the components and reconstructs the image. Deconvolutional layers work similar to upsampling but yield higher performance. Authors explain that if only convolutional layers are used, then the noise will be removed little by little through the layers. During this process, some image details may be eliminated. However, in the proposed algorithm using deconvolutional layers helps to preserve the details.

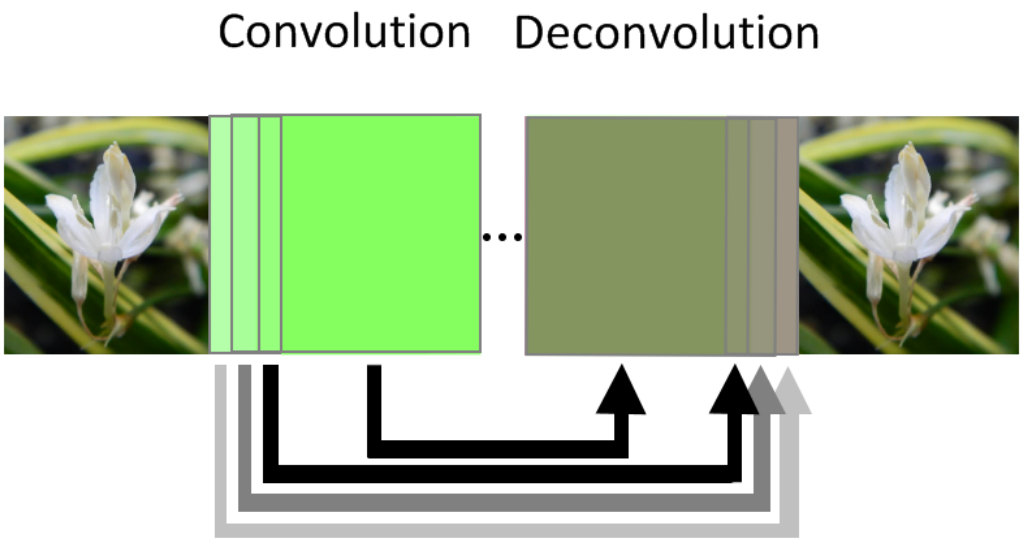


Figure 3.20: Denoising autoencoder architecture

As it can be seen in Figure 3.20 shortcuts are used in this network to pass a feature map from convolutional layers to deconvolutional layers (passes details of the image to upper layers). It also back-propagates the gradients to the bottom layers and makes the training easier.

In high-level applications such as segmentation or classification, pooling is helpful to remove the redundant details of the image. However, in low-level applications like denoising or image super-resolution, it is necessary to preserve all the details and only remove the noise. Therefore, in the second group, it is not common to use pooling layers.

3.8 Dilated Convolution

Most of the networks designed for classification tasks employ pooling or down-sampling layers to reduce the resolution and obtain a global prediction [51, 52]. In dense prediction, the goal is to predict the correct label for each pixel in the image. In such a task, after

finding the global prediction, up-convolutions are used to go back to the original image size and recover the lost resolution [53, 54]. These studies use pooling and down-sampling layers to increase the receptive field and get more contextual information.

Receptive Field (RF) is the region of the input image that is used to calculate the output value. In a convolutional network, the first layer with a 3×3 filter has an RF of 3×3. The second layer with the same size filter grows the RF to 5×5. A large RF contains more contextual information and achieves better results. Down-sampling is a powerful method to increase the RF; however, some detail information are lost by down-sampling that cannot be fully recovered by up-convolution layers. This is especially problematic in tasks such as image restoration, denoising and image super-resolution. Another approach to enlarge the RF without subsampling is using larger filters or more layers. Both of these methods increase the number of weights and computations considerably while the RF grows only linearly.

In 2016, Fisher Yu et al. [55] have used dilated convolution to increase the receptive field without facing the mentioned problems. At the same time, another study employed a similar technique calling it atrous convolution [56]. The main idea behind these studies is the same, and in kinds of literature, the operation is called both dilated convolution and atrous convolution. In this work, we will refer to this operation as dilated convolution. One dimensional dilated convolution is defined as:

$$y[i] = \sum_{k=1}^{K} x[i + r.k]w[k] \tag{3.20}$$

here, $x[i]$ and $y[i]$ are the input and output of the dilated convolution and w represents the weight vector of the filter with length K. The parameter r is the rate of the convolution. If r =1, the dilated convolution will be the standard convolution. Figure 3.21 demonstrates how the dilated convolution is calculated.

The receptive field of each layer can be calculated using Equation 3.21 [57].

$$RF_l = RF_{l-1} + (k - 1)r \tag{3.21}$$

Table 3.1 helps us to investigate how dilated convolution changes the receptive field compare to standard convolution. It can be seen that for a 3-layer dilated convolutional network with $r = 4$ the receptive field is almost double of that for standard convolution.

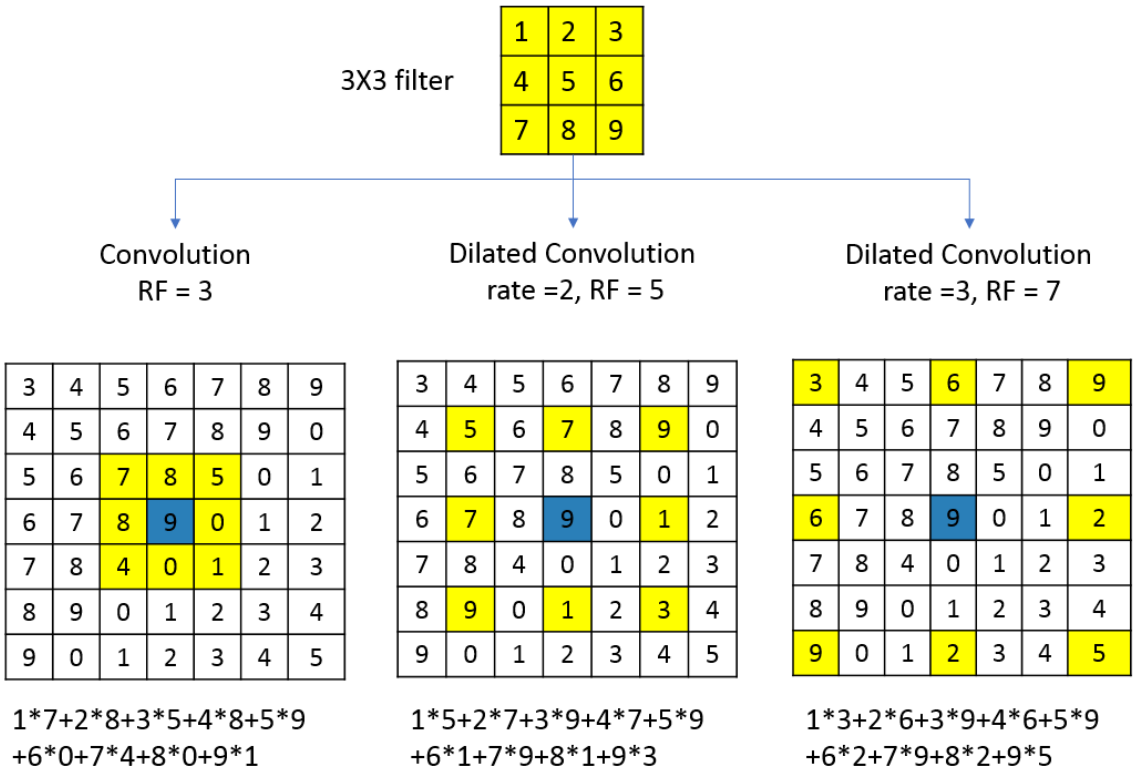


Figure 3.21: Dilated convolution with different rates and the corresponding receptive field (RF).

As discussed before, using dilation convolution will help to achieve a desired receptive field with the fewer number of weights. Equation 3.22 calculates the number of weights in a N layer network with filter size $f \times f$ and n filters in each layer. c represents the number of channels in the input/output image (here, we assume they are same) which is 1 and 3 for grayscale and RGB image, respectively.

$$number\,of\,weights = n \times f^2 \times c + n^2 \times f^2 \times (N-2) + n \times f^2 \times c \tag{3.22}$$

Table 3.2 compares the number of weights needed to achieve receptive field equal to 13 with different dilatation rates. In this example, the filter size is 3×3, and the number of filters in each layer is 64. It can be seen that when dilated convolution is used, we

Table 3.1: Receptive field of a neural network at layers 1 to 3 with dilation rates (r) equal to 1 (standard convolution, 2, 3, and 4, when the filter size is 3×3).

Dilation rate	**r=1**	**r=2**	**r=3**	**r=4**
Layer 1	3	5	7	9
Layer 2	5	7	9	11
Layer 3	7	9	11	13

Table 3.2: Number of weights needed for $RF = 13$ with different dilation rate and 3×3 filter.

Dilation rate	$r = 1$	$r = 2$	$r = 3$
Number of layers needed for $RF = 13$	6	5	4
Number of weights	148,608	111,744	74880

can achieve a large receptive field with a fraction of weights that are needed when using standard convolution($r = 1$).

Besides adding more layers to grow the receptive field, it is possible to use larger filters and reach the desired receptive field in fewer layers. However, using larger filters increases the number of weights drastically, as it can be seen in table 3.3. Tables 3.2 and 3.3 clearly show that for a specific receptive field, dilated convolution needs a fraction of the weights compared to using more layers or larger filters.

Dilated convolution was initially used in dense prediction and semantic segmentation tasks [55, 56]. Later, researchers have used it for different purposes with successful results. Dilated convolution has shown great potential to remove noise from noisy images [58, 57].

The power of dilated convolution is that it does not compel any changes to the network and can replace standard convolution in any network and grow the receptive field. To understand how dilated convolution can improve the performance of a network, Wang

Table 3.3: Number of weights needed for $RF = 13$ with different filter sizes.

Filter size	3×3 $r = 1$	5×5 $r = 1$	7×7 $r = 1$	3×3 $r = 3$
Number of layers needed for RF=13	6	5	4	4
Number of weights	148,608	310400	407680	74880

et al. [57] have replaced the standard convolution by dilated convolution in the DnCNN network proposed by Zhang et al. [47]. DnCNN (Figure 3.19) is a deep convolutional network designed to remove noise from natural images consisting of 17 layers for grey images and 20 layers for colour images. Wang et al. argue that if dilated convolutions with a rate of two are used in the middle layers in DnCNN then the same receptive field can be obtained with comparable performance in just 10 and 12 layers for grey and colour images, respectively. As a result, the number of parameters, the needed memory, and the training time is almost half for their network compared to DnCNN.

Deep neural networks as a branch of artificial intelligence have made some problematic tasks more manageable. Similar to a human, it learns from observing samples and distinguishes patterns in the data. The enormous computational resources allow to train deep networks and work with massive data. New advancements such as residual learning, batch normalization have provided better and easier optimization. This study employs these techniques to enhance the quality of low dose CT images.

Chapter 4

Proposed Methodologies

4.1 Network Architecture

In this study, we have proposed two deep neural networks to remove noise from low dose CT images. The first network is a Dilated Residual Learning (DRL) [3]. This network outperforms BM3D [2], two deep networks, CNN200 [30] and Zhang [58]. Later, we improved the performance of the network by adding an edge detection layer and employing a combination of per-pixel loss and perceptual loss functions for the objective function.

4.2 Dilated Residual Learning (DRL)

The proposed network architecture is shown in figure 4.1. The network has seven layers and employs three deep learning techniques: batch normalization, residual learning, and dilated convolution.

4.2.1 Batch Normalization

As explained in the previous chapter, batch normalization (BN) was proposed to solve the exploding/vanishing gradient problems. This technique normalizes the input of activation functions for each batch of training data. In the proposed network, batch normalization is used in layers 2 to 6 before ReLU. By using BN, the network converges much faster.

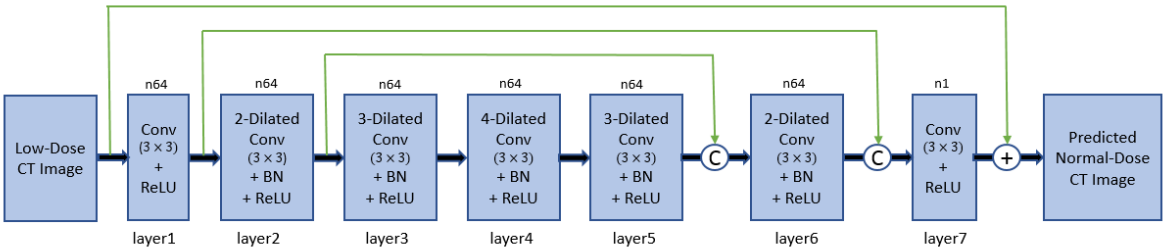


Figure 4.1: Architecture of the diated residual network (DRL)

It also solves the problem of parameter initialization as the initialization will have less impact on the final results.

4.2.2 Residual Learning

The initial idea of the proposed network was inspired from a study by Zhang et al. [58]. In the research, a seven-layer dilated convolutional network is used for image restoration. When we applied low-dose CT images to their proposed network, the output showed some improvement. However, we noticed that the network does not take advantage of residual learning at its full potential. Our experiments demonstrated that combining the residual learning with the network can boost the performance. For this purpose, we added the skip connections between symmetric layers. In our network, the output of layers 1 and 2 are concatenated with the output of layers 6 and 5, respectively. The feature maps obtained from the first layers contain more details from the image. Through skip connection, we pass this information to the higher layers (layers 5 and 6). In our network, each layer has 64 filters, so layers 5 and 6 receive 64 feature maps from their previous layers (layers 4 and 5) and also 64 feature maps from the first layers (layers 2 and 1, respectively).

This architecture improves the performance as layers 5 and 6 have access to both processed and low-level information. There is another skip connection in our network that sends the input, low-dose CT image, directly to the output of the last convolutional layer. However, this time we perform arithmetic addition between them, not concatenation. The final result achieved after this addition is the clean image. Therefore, we can consider that this seven-layer convolutional network finds the inverse of noise in the low-dose CT

Table 4.1: Receptive field of the proposed network for a 3×3 filter

layer 1 dr=1	layer 2 dr=2	layer 3 dr=3	layer 4 dr=4	layer 5 dr=3	layer 6 dr=2	layer 7 dr=1
3	7	13	21	27	31	33

Table 4.2: Receptive field of a 7-layer network with standard convolution for a 3×3 filter

layer 1	layer 2	layer 3	layer 4	layer 5	layer 6	layer 7
3	5	7	9	11	13	15

image and by adding it to the noisy low-dose CT image, we can recover the normal-dose image.

4.2.3 Dilated Convolution

As explained before, dilated convolution makes it possible to increase the receptive field without adding more layers or using larger filters. Larger receptive field means that the convolutional layer can look at the larger area from the image and capture more contextual information. In our proposed network, we have used dilated convolution in the all the layers except fist and last one. The rates of dilation for these layers are different and as we get closer to the center of the network, the rate increases. If we consider, the standard convolution as a dilated convolution with rate 1, the dilation rates are 1, 2, 3, 4, 3, 2, and 1 for layers 1 to 7, respectively. As table 4.1 demonstrates the receptive field of this network for the filter size 3×3 is 33.

Table 4.2 shows the receptive field of a 7-layer network that uses standard convolution. By comparing these 2 tables we can see that the employing dilated convolution in our network increased the receptive field more than 2 times. This growth is achieved without adding more layers or using larger filters.

4.2.4 Objective

Let X be the noisy low-dose image and Y denote the corresponding normal-dose CT image. The goal is to find $f(X)$ that is as close as possible to Y. The objective function

$\mathcal{L}$ is mean square error (MSE) and is defined as follow:

$$\mathcal{L}(\Theta) = ||f(X) - Y||_F^2 \tag{4.1}$$

In deep learning, similar to other machine learning techniques, we train the network with providing many samples. Therefore the above objective function can be written as bellows:

$$\mathcal{L}(\Theta) = \frac{1}{N}\sum_{i=1}^{N} ||f(x_i;\Theta) - y_i||_F^2 \tag{4.2}$$

In this equation, N is the number of samples and (x_i, y_i) is a pair of low-dose, normal-dose CT image. Θ represents the set of parameters that should be learned during the training process.

It is possible to use other objective functions such as Structural Dissimilarity (DSSIM), but in most of the studies done for image noise removal, image enhancement, and image super-resolution, mean square error is used. We have trained our network with both SSIM and MSE but comparing the result showed when MSE is employed as the objective function that performance is better. Therefore, we chose to use mean square error as the objective function.

4.3 DRL with Edge Detection Layer

Although dilated residual network delivered good results, we noticed that some of the details in the image are not clearly visible in the output image. To improve the performance of DRL, we have introduced an edge detection layer that helps to extract edges and increase the visibility of the details in the output. Figure 4.2 displays Dilated Residual Learning with Edge Detection Layer (DRL-E). Another difference between this network and DRL is the choice of objective function. During our experiments, we observed that images produced by the network are blurred and over smoothed which is the result of optimizing by mean square error. Studies in super-resolution tasks have also noticed the similar problem [59, 60]. However, it is more critical in our case as the results will be used for making a diagnosis. To better detect the abnormalities in the organs, physicians generally use applications such as Dicom Viewer that allows them to examine the CT

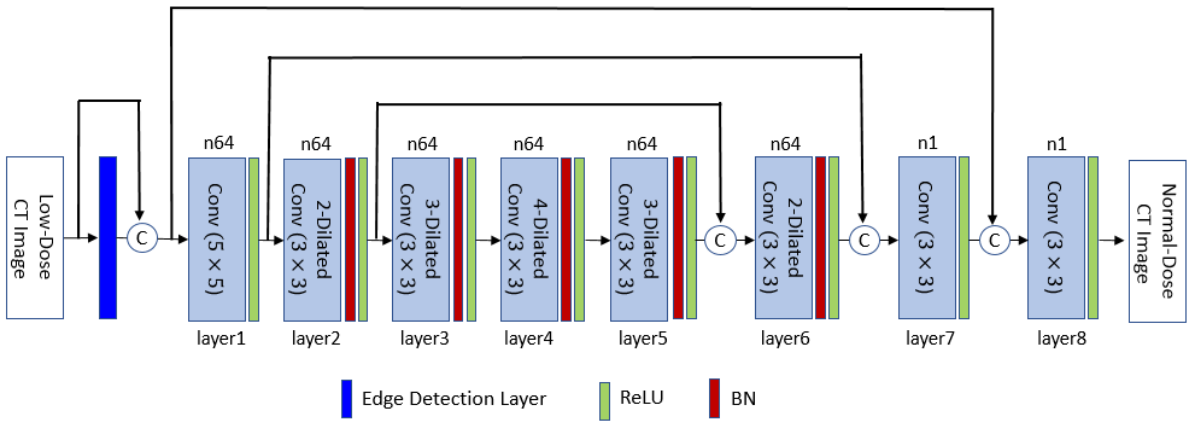


Figure 4.2: Architecture of the diated residual network with edge detection layer

image with different contrasts and gray level mappings. The process is called windowing, and it helps to high light the appearance of different structures. We have remarked that the over-smoothing problem is more pronounced in some windows such as abdomen window that expose more texture details. We have solved this problem by employing an objective function that joins the perceptual loss and MSE loss.

4.3.1 Edge detection layer

Edge detection has been widely used in many image processing and computer vision tasks. The goal of edge detection is to extract the boundaries of the objects within the image. Different techniques have been proposed to perform this task, and they mostly search for discontinuities in the image brightness. Sobel edge detection is a simple and popular algorithm that computes the 2-D gradient of the image intensity by convolving 3×3 kernels with it. The algorithm emphasizes the regions with high spatial frequency and extracts the edges.

In this study, we have proposed an edge detection layer to improve the performance of the DRL network. This layer adapts the Sobel algorithm to detect edges in vertical, horizontal and diagonal directions. The layer does not have any trainable parameters and therefore do not add to the complexity of the network. The edge detection layer is a regular convolutional layer with four predefined filters with no activation function. Figure 4.3 displays Sobel kernels that are used as predefined filters in the convolution. In

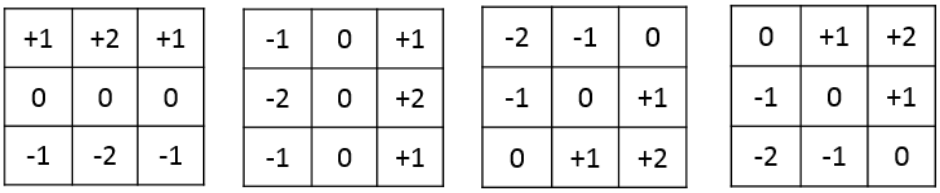


Figure 4.3: Sobel edge detection kernels, (a) Horizontal direction, (b) Vertical directin (c) 45° diagonal direction, (d) 135° diagonal direction

the proposed network, the output of this layer is concatenated with the input image and forms a data with five channels. As Figure 4.2 shows, it is sent to the layer 1 and layer 7 of the DRL network. Our experiments confirmed that this layer enhances the output result.

4.3.2 Objective Function

Mean square error is the most common loss function in statistics and is generally used to optimize neural network designed for super-resolution or noise removal. It minimizes the per-pixel difference between the results and the ground truth. Optimizing by MSE leads to achieve higher Peak Signal to Noise Ratio (PSNR); however, it does not guaranty that the final result is visually appealing. During our experiments, we faced this problem. DRL-E network produced better results compared to the other networks, though the output images suffered from over-smoothing and some of the details were lost.

Johnson [59] et al. have showed that perceptual loss could considerably enhance the outcome. The perceptual loss is computed by comparing the feature maps generated by a pre-trained neural network. VGG16 [61] is the network that is often used to create the feature maps. It was designed for image recognition and trained on the Imagenet dataset [62]. Figure 4.4 displays this network. To measure the perceptual loss, an image is given to the network as an input, and the feature maps from one or multiple blocks are extracted. The perceptual loss is calculated by finding the means square error between the feature maps obtained for the image and the ground truth. In this study, we have used four group of feature maps from blocks 1, 2, 3, and 4. The feature maps are extracted from the last convolutional layer in these blocks and before the pooling layer. Figure 4.5

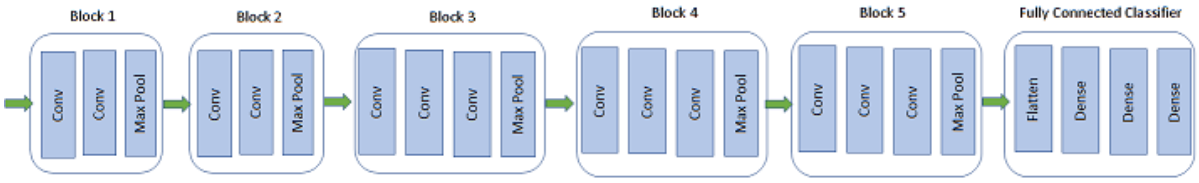


Figure 4.4: VGG16 network designed for image recognition)

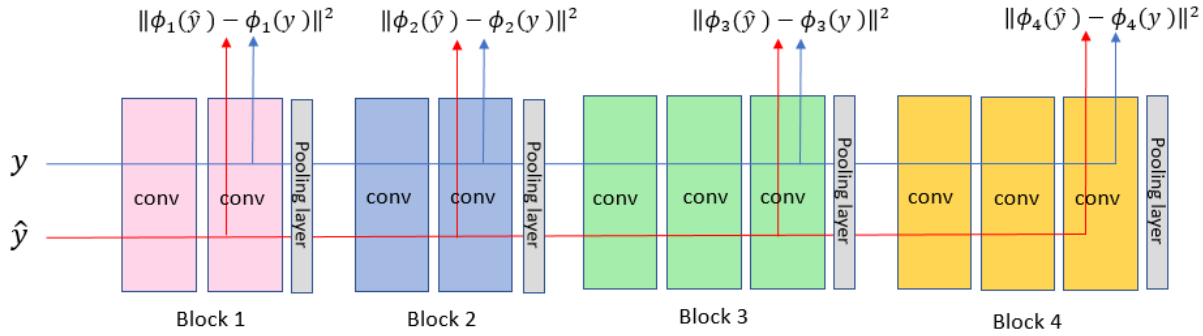


Figure 4.5: Perceptual loss is computed by extracting the feature maps of blocks 1, 2, 3, and 4 from a pre-trained VGG-16 network.

shows these feature maps. The perceptual loss function $\mathcal{L}_P(\theta)$ is as follows,

$$\mathcal{L}_P(\theta) = \sum_{i=1}^{4} \mathcal{L}_i(\theta) \tag{4.3}$$

$$\mathcal{L}_i(\theta) = \frac{1}{h_i w_i d_i} ||\phi_i(\hat{y}(\theta)) - \phi_i(y)||^2 \tag{4.4}$$

here, y and $\hat{y}$ represent the ground truth and denoised image from the network, and $\phi_i(\hat{y})$ refers to the extracted feature maps from block i with size $h_i \times w_i \times d_i$.

Training the DRL-E network with perceptual loss helps to preserve many structural details; nevertheless, it results in grid-like artifacts in the output image. To reduce these artifacts and achieve a better outcome, we have used a combination of MSE loss and perceptual loss. The objective function used to optimize the network is as follows,

$$\mathcal{L}(\theta) = \lambda_{mse}\mathcal{L}_{mse}(\theta) + \lambda_P\mathcal{L}_P(\theta)\lambda_{mse} + \lambda_P = 1 \tag{4.5}$$

where, λ_{mse} and λ_P are weighting scalars for mean-square error loss $\mathcal{L}_{mse}$ and perceptual loss $\mathcal{L}_P$, respectively.

The experiments exhibited that utilizing the above objective function to train DRL-E network helps to resolve both blurring and grid-like artifacts problems.

In this study, we have searched for the techniques to improve the performance of the network for a fixed number of layers. It is clear that stacking more layers will improve the performance; however, it inflates the number of the parameters and therefore the complexity of the network. Dilated convolution is a powerful tool to achieve this goal. The proposed edge detection layer can be added to any network to obtain sharper edges with little change in the number of weights. Moreover, the objective function 4.5 allows preserving the textural details and enhance the outcome.

Chapter 5

Dataset Preparation

Low-dose CT scan exposes a patient to less radiation compared to normal-dose CT scan. However, the obtained images are noisy and distorted. We have proposed a deep neural network that maps the low-dose CT images to the clean images. Training a neural network, similar to other machine learning algorithms, needs many training samples. The larger dataset with different samples provides better generalization.

To obtain good results from a deep neural network, it is critical to train the network with many samples. In our case, we needed low-dose and normal-dose CT image pairs to use as data and label. However, the available datasets are limited. To tackle this problem, we have employed a simulation algorithm to generate low-dose CT images from normal-dose images and use the pair to train the network.

5.1 Low Dose CT Image Simulation

In X-ray radiography, the number of photons that leave the source and the number of those photons that are captured by the detector can be modeled by Poisson distribution[63]. The poisson distribution looks at the probability of an event occurring in a given range of time, where the event is the number of photons detected within that time. There are situations where Poisson distribution does not hold such as when it is not possible to detect photon count due to high flux [63]. Instead of the detector counting the photons, it acts as an integrator to obtain an average current [63]. For cases where the sinogram does not hold Poisson distribution, a Noise-Equivalent Count (NEC) scaling or shifting

is used in order to approximate Poisson noise [63]. NEC rate is a widely used indicator for the Signal to Noise Ratio (SNR) having proportionality in its relationship [64]. A difference is that NECR analyzes the signal to raw noise data while SNR analyzes the signal to the reconstructed image noise [65].

According to the works of literature, the dominant noise in X-ray and CT scan imaging has Poisson distribution [66, 67]. Therefore, to simulate a low-dose CT image, we can add Poisson noise to the sinogram of the normal-dose image.

5.1.1 Preprocessing

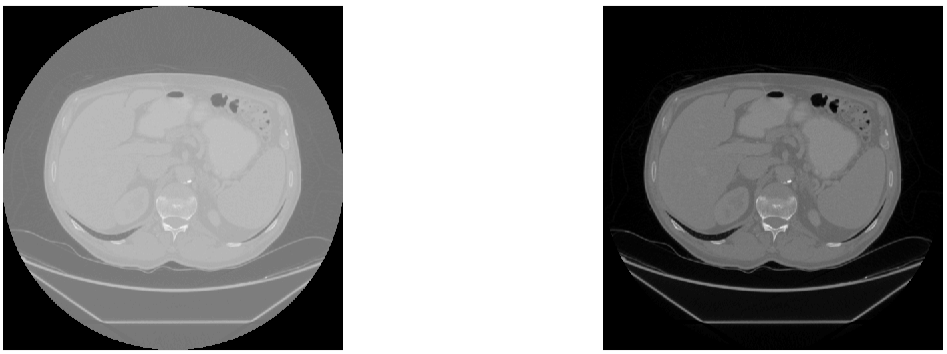
To create our dataset, we have used an algorithm to simulates the low dosage CT images from normal dosage by adding Poisson noise to the projection data [68]. The low dose CT images obtained from the program will be used for deep learning to train an algorithm to reconstruct normal dose CT images from low dose CT images. Before adding Poisson noise to the CT images, the data will be formatted by applying the required transformations.

To prepare the data, the pixel values from the CT images need to be converted to its attenuation coefficients. In order to convert the values accordingly, first, we find the pixel values by using the Digital Imaging and Communications in Medicine (DICOM) package. Then, we remove the padding by finding the padding value from the metadata of the dicom image and then replace pixels values with such an amount by zero [69]. Usually, the padding is a value of -2000. Figure 5.1b displays the result of padding removal.

In the next step, we apply the linear transformation 5.1 to obtain Hounsfield units based on the current pixel value.

$$HU = \frac{PixelValue}{Slope} + Intercept \tag{5.1}$$

The slope and intercept can be accessed through the metadata of the Dicom images. Referencing Table 2.1, the higher HU portions of tissue have higher intensity such as bone (bone $>$ 700 HU) compared to the space surrounding the patient (air $= -1000$ HU) being low intensity. Most soft tissues will be $<$ 500 HU due to larger attenuation. This scale gradient helps contrast the tissue being observed and can indicate whether



(a) (b)

Figure 5.1: CT scan image padding removal (a) Lung CT image with padding, (b) Lung CT image with after padding removal

there are discrepancies in comparison to healthy organs such as blood clots or tumours.

5.1.2 Noise simulation algorithm

As explained before, to produce low-dose CT images, we need to add noise to the projection data. Therefore, all the transform applied to generate a CT image from the projection data should be inverted. To recover the linear attenuation coefficients from Hounsfield numbers, we apply the inverse of Equation 2.2. Equation 5.2 performs this transform.

$$\mu_{nd} = \frac{\mu_{water}}{1000} HU + \mu_{water} \tag{5.2}$$

Next, the linear attenuation values are converted to projection data (sinogram data) using radon transform. Figure 5.2 displays the sinogram of the CT image in Figure 5.1. The sinogram is multiplied by the voxel size in order to eliminate size factor [70]. Voxel size is stored in metadata item pixel spacing or can be calculated by dividing the Reconstruction Diameter (from metadata) to 512 (number of pixels in the image).

$$\rho_{nd} = radon(\mu_{nd}) \times voxel \tag{5.3}$$

where, ρ_{nd} represents projection or sinogram data for the normal-dose CT image.

Transmission of normal dose data (T_{nd}) is calculated from the sinogram [68].

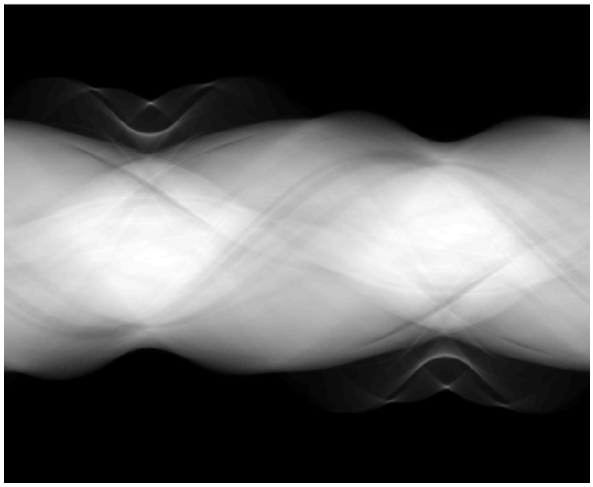


Figure 5.2: Sinogram of the lung CT image in Figure 5.1b

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

Then, the low dose transmission can be generated using Poisson noise [68].

$$T_{ld} = Poisson(I_{ld}^{o} T_{nd}) \tag{5.4}$$

here, I_{ld}^{0} is simulated low-dose scan incident flux and T_{ld} is low-dose transmission data.

Next, the low dose sinogram is generated by using the low dose transmission from equation 5.4.

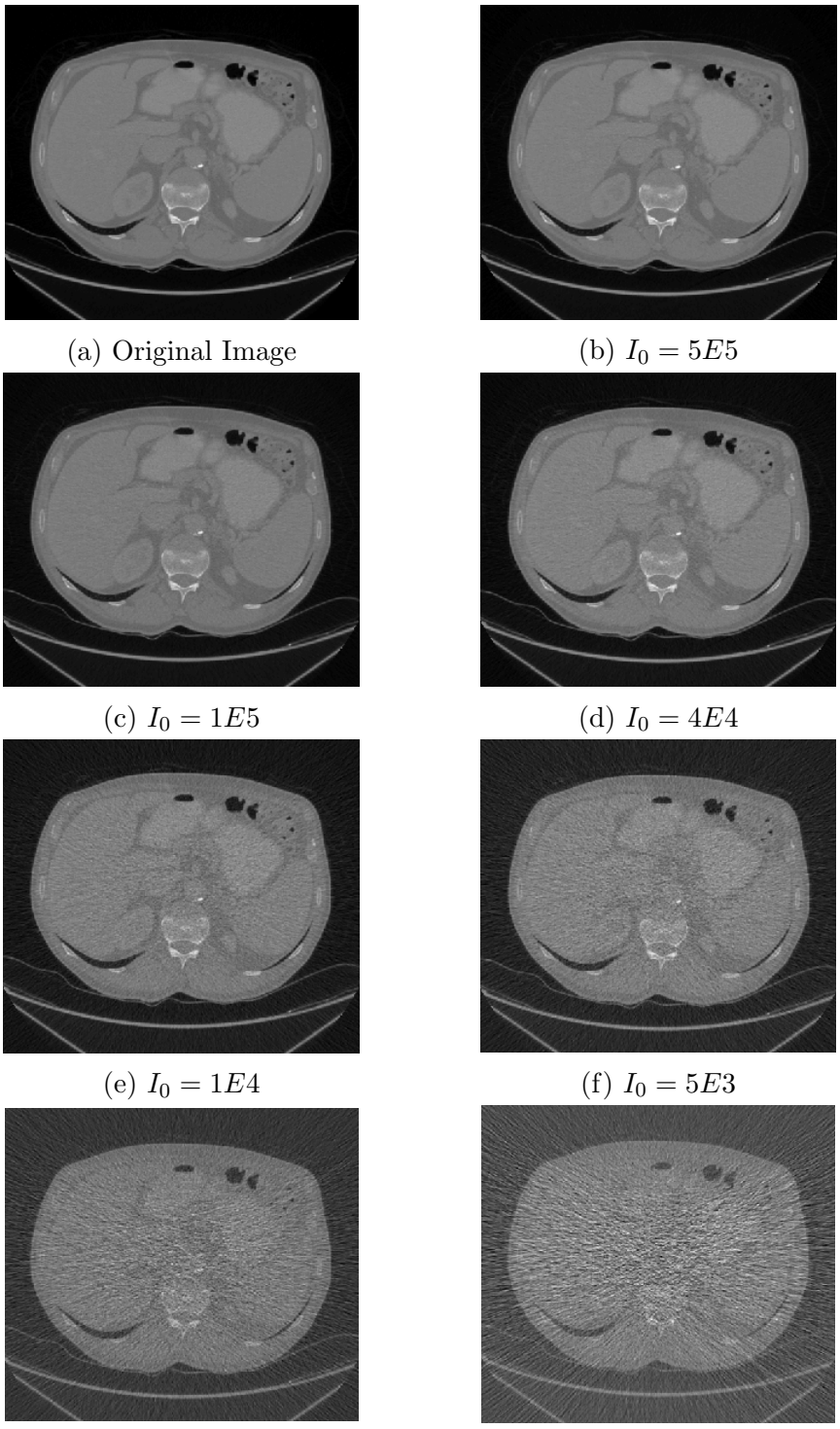
$$\rho_{ld} = ln(\frac{I_{ld}^{o}}{T_{ld}})$$

To obtain a better low-dose CT image, we use the difference between the normal dose and low dose sinograms to get the noise projection [70].

$$\rho_{noise} = \rho_{nd} - \rho_{ld}$$

Then the inverse Radon transform is applied to revert the noise projection to the linear attenuation of CT image which then is added to the normal dose CT image.

$$\mu_{ld} = \mu_{nd} + iradon(\frac{\rho_{noise}}{voxel})$$



(a) Original Image

(b) $I_0 = 5E5$

(c) $I_0 = 1E5$

(d) $I_0 = 4E4$

(e) $I_0 = 1E4$

(f) $I_0 = 5E3$

(g) $I_0 = 2E3$

(h) $I_0 = 1E3$

Figure 5.3: Simulation of low-dose CT images

To generate Hounsfield numbers for the low-dose CT image, the inverse of Equations 5.2 and 5.1 should be applied.

With this algorithm, it is possible to create low-dose datasets with different X-ray currents. These images will be fed to the deep learning network and train the system to generate normal-dose CT images from low-dose CT images. The variety of noise levels ensure that the system has experience with different amounts of noise, which can be varied based on dosage levels, patient misalignment, filters, and CT scanner parameters and equipment. Figure 5.3 displays the output of this system for different X-ray currents.

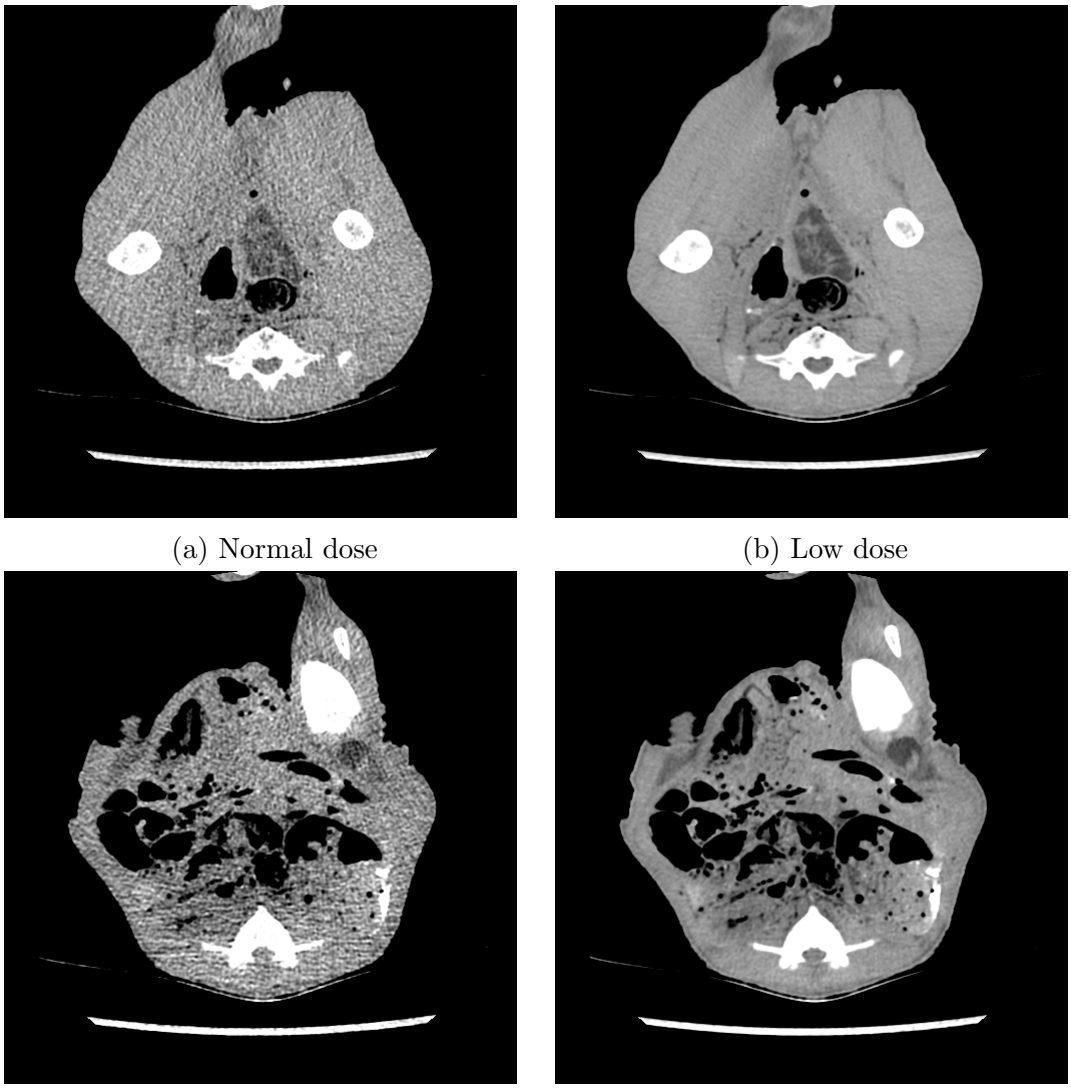
To generate simulated low-dose dataset for our exxperiments, we downloaded lung CT scans for a patient [71] from The Cancer Imaging Archive (TCIA) [72] and simulated low dose CT images by the method explained in section 5.1. The original dataset included 663 images and with Current X-ray tube of $330mAs$, the peak voltage of $120KVp$ and slice thickness of $1.25mm$. We have used the incident flux (I_{ld}^0) equal to 2×10^3 in equation 5.4 to generate simulated low dose CT.

5.2 Real Piglet Dataset

The second dataset is taken from a deceased piglet. The CT scans are acquired with $300mA$ and $15mA$ X-ray currents at entirely similar conditions to produce normal-dose and low-dose CT scans. The dataset includes 900 slices with $100KVp$ peak voltage and $0.625mm$ thickness. Therefore, the low-dose CT images have been aqcuired 5% the X-ray current compared to the normal-dose ones. Figure 5.4 displays a few images from this dataset.

5.3 Phantom Thoracic Dataset

The last dataset that is used in this study are acquired from an anthropomorphic thoracic phantom containing a vasculature insert to which synthetic nodules were inserted or attached [73]. We refer to this dataset as the Thoracic dataset. The dataset has 407 pairs of images with the peak voltage of $120KVp$ and slice thickness of $0.75mm$. The X-ray tube current for normal-dose and low-dose CT images are $480mAs$ and $60mAs$, respectively. Figure 5.5 shows a few images from this dataset.



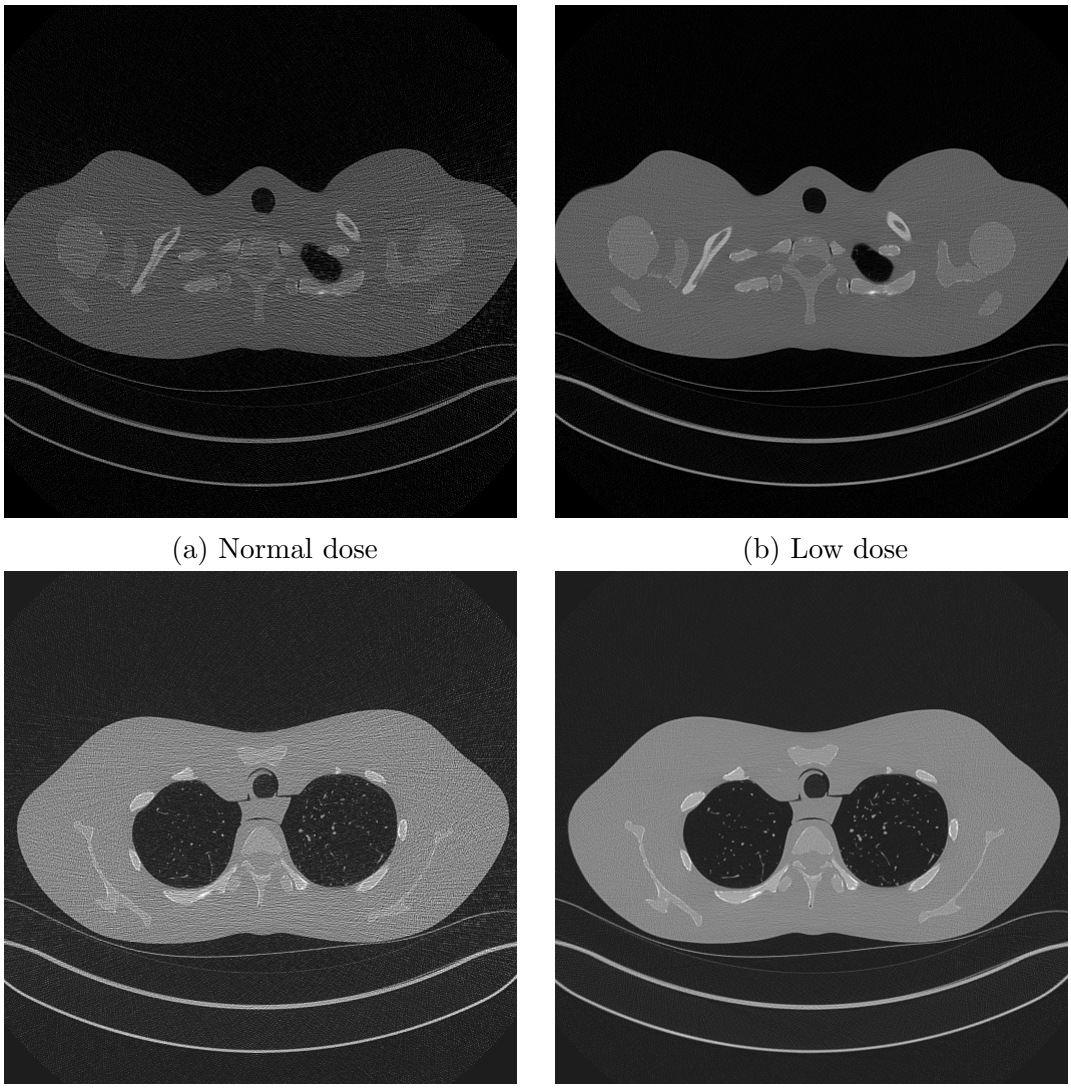
(a) Normal dose

(b) Low dose

(c) Normal dose

(d) Normal dose

Figure 5.4: CT images from Piglet dataset



(a) Normal dose (b) Low dose

(c) Normal dose (d) Normal dose

Figure 5.5: CT images from Thoracic dataset

Chapter 6

Result and Discussions

6.1 Experiments Setup

To examine the performance of the proposed networks, we have used three datasets as explained in previous chapter.

To prepare the data for training the networks, we have used the pixel values of each CT slice and divided it by 4095. In this way, the data is mapped between 0 and 1 which is the recommended range for training neural networks. We have also extracted 40×40 patches with the stride of 20 pixels from the CT images. The original size of a CT image is 512×512 and patch extraction helps to boost the number of training data. Moreover, it makes training easier on a system with no big memory. We have chosen to crop patches to 40×40 because the receptive field of the DRL-E network is $5 + 4 + 6 + 8 + 6 + 4 + 2 + 2 = 37$ in each direction. Since the proposed network is fully convolutional, it is size independent meaning the input image can have any size. The test images with size 512×512 can be fed to this network without any alteration.

In this research, the original dataset is split to 70% and 30% to build training and test datasets, respectively. In many studies the test images are chosen randomly from the primary dataset; however, here, the last 30% of CT images are held for the test dataset. The reason is that the consecutive CT images are generally similar to each other, as the slice increment is minimal. Slice Increment is the movement of the table/scanner for scanning the next slice. Testing the network on the last 30% segment exhibits how the network will perform on new data.

The activation function used in the convolutional layers is rectified linear unit (ReLU), and zero-padding is used for convolutions to avoid boundary artifacts [58]. There are 64 filters in layers 1 to 6 in both networks, while layer 7 in DRL and layers 7 and 8 in DRL-E have just one filter.
Five networks are trained in these experiments with similar conditions to ensure that the improvements are accomplished because of the modifications, not different training. The weights are initialized by Glorot normal [74] and the learning rate for the first 20 epochs is $1e-3$, and then it is reduced to $1e-4$ for the next 20 more epochs.

The networks are implementd on Keras with Tensorflow backend on a system with an Intel core $i7$ CPU $3.4GHz$, $32G$ memory and GeForce GTX 1070 Graphics Card.

To evaluate the performance of the proposed networks, the state of the art BM3D algorithm [2] as a traditional image denoising method, and neural networks CNN200 [30] are selected. Also, we have made a comparison on the network proposed in Zhang [58] which was the inspiration behind the networks in this study.

We have also trained and tested 4 more networks to assess the effectiveness of each modification: The first one is the dilated residual learning (DRL) (Figure 4.1) that examine how adding residual learning helps to gain better results. The weights of this network are learned by optimizing the Mean Square Error (MSE). The second network is dilated residual learning with edge detection layer (DRL-E)(Figure 4.2), which is trained by three different objective functions. The first training is performed by MSE loss function, and we refer to it as DRL-E Network Optimized by MSE (DRL-E-M). The comparison between this network and DRL determines that adding the edge detection layer is an efficient method to enhance the outcome of the network. The second training on DRL-E is done by minimizing the perceptual loss defined in Equation 4.4 and is called DRL-E Network Optimized by Perceptual Loss (DRL-E-P). The last training is DRL-E Network Optimized by MSE and Perceptual loss (DRL-E-MP)as explained in Equation 4.5.

6.2 Results

6.2.1 Simulated Lung Dataset

The first experiment performs denoising on the simulated low-dose Lung dataset. The networks offer an end-to-end solution to the problem, so the low-dose CT images are

Table 6.1: The average PSNR and SSIM of the different algorithms for the Lung dataset.

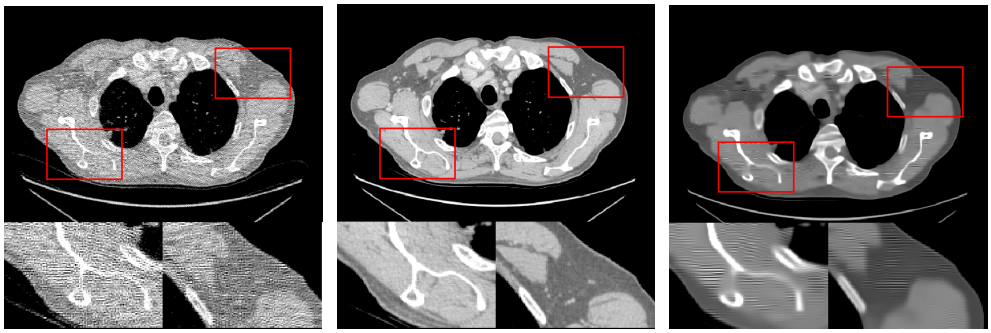
Metric	Low-dose image	BM3D	CNN200 [30]	Zhang [58]
PSNR	14.59	24.76	33.19	33.74
SSIM	0.2008	0.6750	0.8768	0.8804

Metric	DRL [3]	DRL-E-M	DRL-E-P	DRL-E-MP
PSNR	34.17	36.64	33.47	35.57
SSIM	0.9281	0.9733	0.5880	0.6910

applied to the input of the networks, and the predicted normal-dose CT images are collected in the output of the networks. In this experiment, seven algorithms are compared. Table 6.1 demonstrate the peak signal to noise ratio (PSNR), and the Structural Similarity (SSIM) achieved for BM3D, CNN200, Zhang, DRL, DRL-E-M, DRL-E-P, and DRL-E-MP. This table shows adding shortcut connections to Zhang [58] improves the performance. Also, employing the edge detection layer with MSE loss function increases both PSNR and SSIM. These results are also perceived in Figure 6.1 and Figure 6.2 showing the outcomes in abdomen window and lung window, respectively. As one can expect, utilizing the perceptual loss does not improve the PSNR. Optimizing by MSE always provides the best PSNR since it looks for parameters that minimize the per-pixel loss. The perceptual loss can better capture the texture details as Figure 6.1 shows. However, it adds grid-like artifacts to the image. DRL-E-MP displays the result of training the DRL-E network with both MSE and perceptual loss which solves the artifact problem while preserving the structural details.

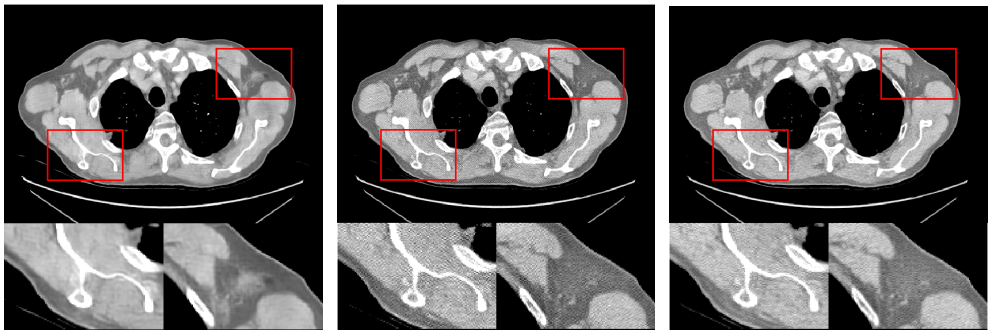
6.2.2 Real Piglet Dataset

The quantitative measures for the Piglet dataset are shown in Table 6.2. Figure 6.3 and 6.4 demonstrate the visual comparison among the seven algorithms. These results are in parallel with observations over the Lung dataset. The performance of the state of the



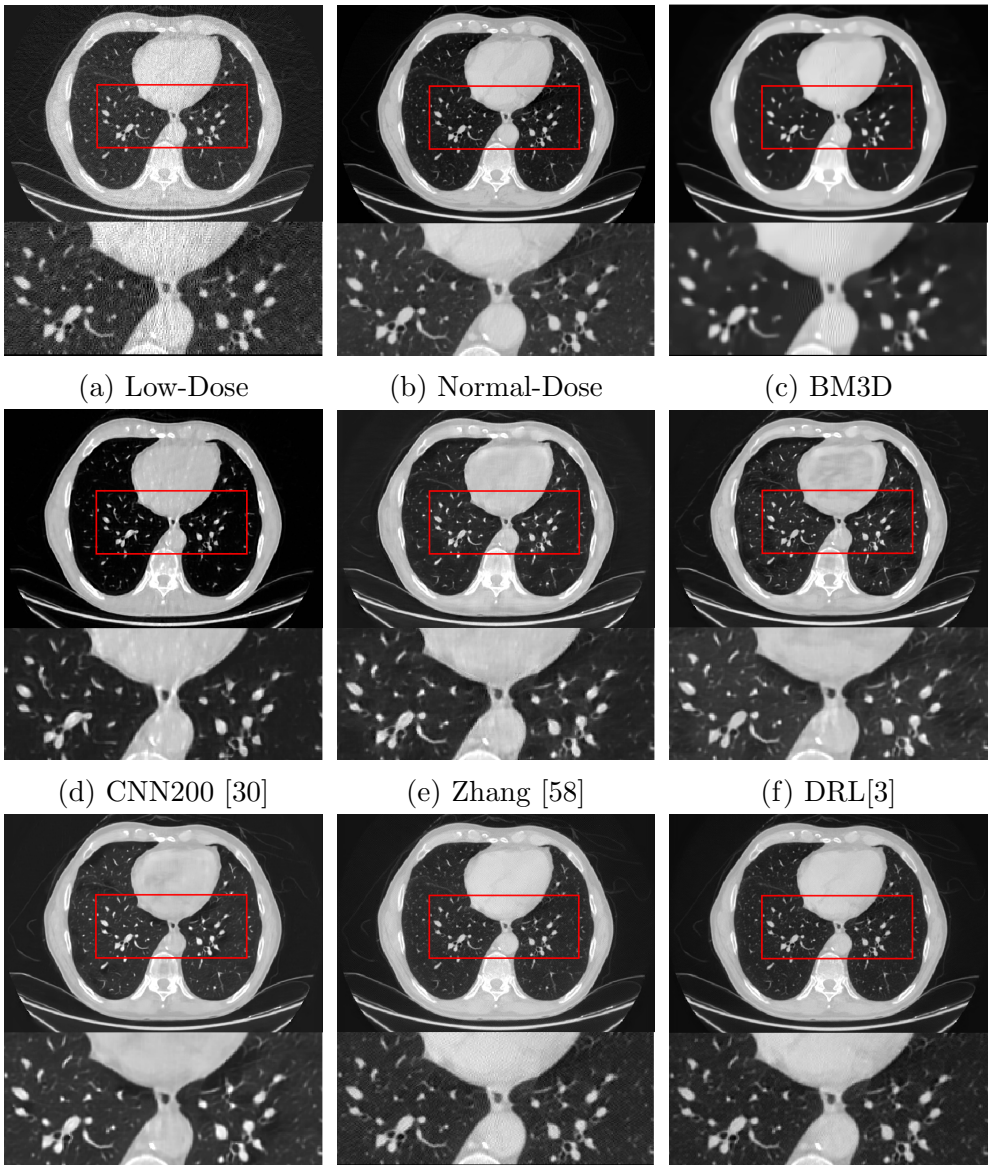
(a) Low-Dose (b) Normal-Dose (c) BM3D

(d) CNN200 [30] (e) Zhang [58] (f) DRL[3]



(g) DRL-E-M (h) DRL-E-P (i) DRL-E-MP

Figure 6.1: Denoising results of the different algorithms on Lung dataset in abdomen window.



(a) Low-Dose (b) Normal-Dose (c) BM3D

(d) CNN200 [30] (e) Zhang [58] (f) DRL[3]

(g) DRL-E-M (h) DRL-E-P (i) DRL-E-MP

Figure 6.2: Denoising results of the different algorithms on Lung dataset in lung window.

Table 6.2: The average PSNR and SSIM of the different algorithms for the Piglet dataset.

Metric	Low-dose image	BM3D	CNN200 [30]	Zhang [58]
PSNR	39.93	41.46	44.18	44.83
SSIM	0.9705	0.9733	0.9804	0.9816

Metric	DRL [3]	DRL-E-M	DRL-E-P	DRL-E-MP
PSNR	44.96	45.10	44.01	44.12
SSIM	0.9881	0.9885	0.9782	0.9807

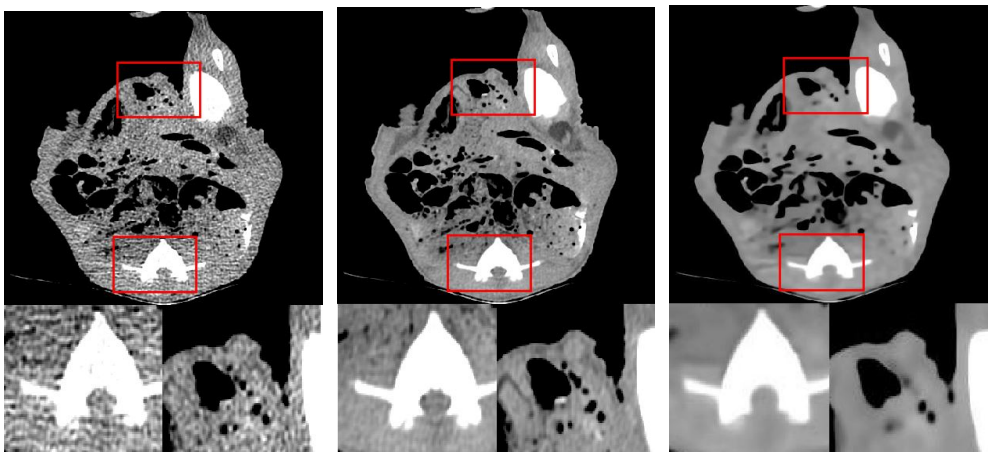
art BM3D is lower than the neural networks. Exploiting perceptual loss in combination with mean-square error removes noise better than other algorithms and generates images similar to the target.

6.2.3 Thoracic Dataset

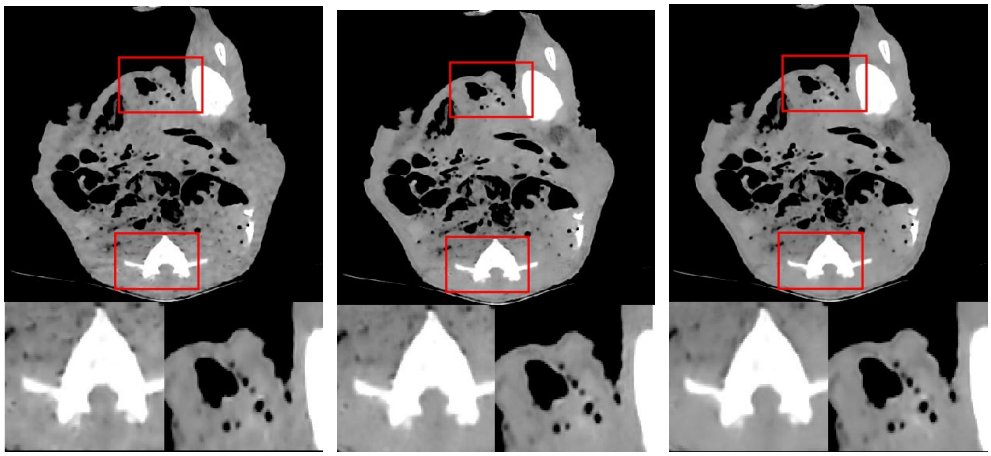
This dataset clearly exhibits the effects of each modification on the network. The PSNR and SSIM values for the algorithms are listed in Table 6.3. Figure 6.5 and 6.6 reveal that while the CNN200 and Zhang [58] achieve some level of denoising, DRL network produces a more clear outcome. Adding the edge detection layer to the DRL leads to sharper and more distinct edges. Comparing the output images for DRL-E-M and DRL-E-P reveals that MSE creates smoother edges and softens the texture. The perceptual loss follows the composition of the target more precisely.

6.2.4 Denoising results on phantom Thoracic dataset

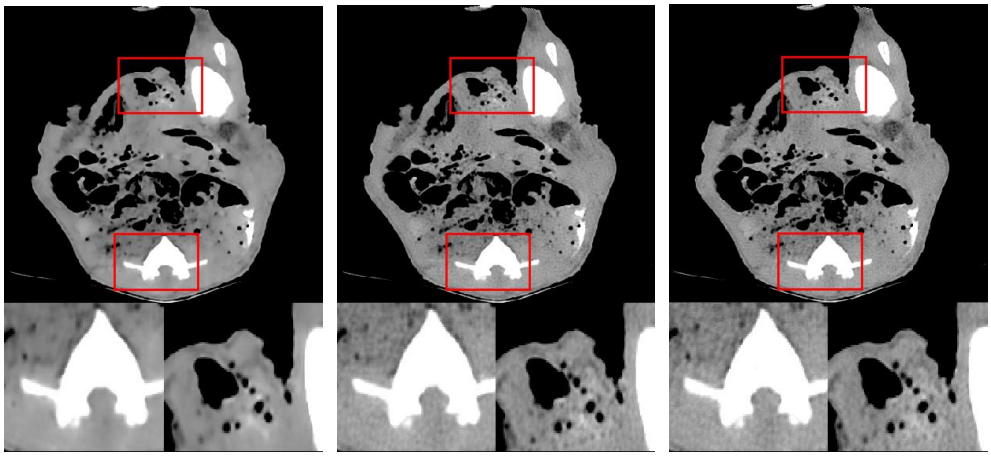
Table 6.3 represents the PSNR and SSIM of denoising Thoracic dataset by all the methods. Results obtained for this dataset is consistent with the other experiments. Figure 6.5 clearly exhibits the effects of each alteration. Comparing the results obtained by DRL and DRL-E-M confirms that the edge detection layer helps to deliver sharper and more precise edges. As explained before, the only difference between these two models is using



(a) Low-Dose (b) Normal-Dose (c) BM3D

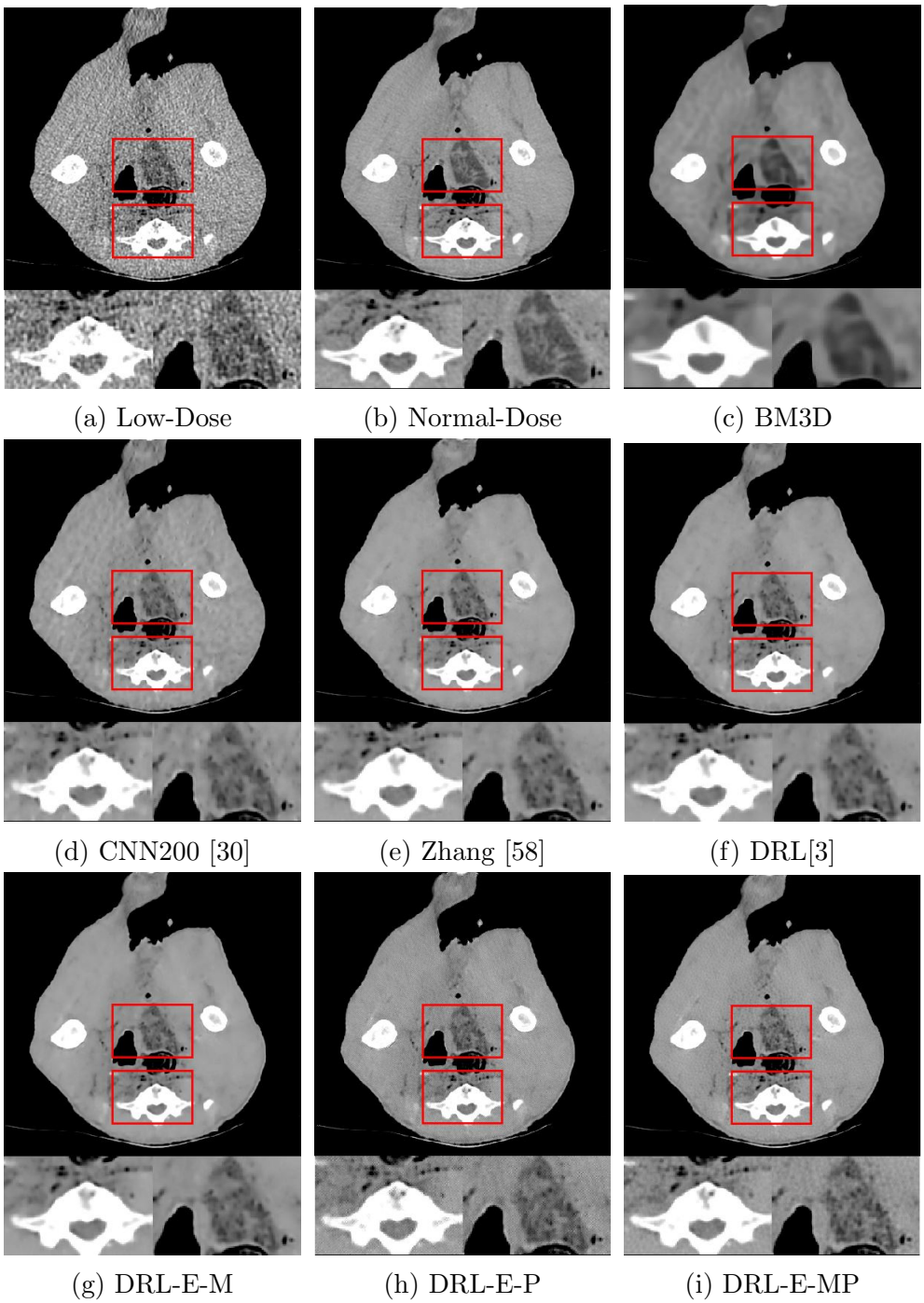


(d) CNN200 [30] (e) Zhang [58] (f) DRL[3]



(g) DRL-E-M (h) DRL-E-P (i) DRL-E-MP

Figure 6.3: Denoising results of the different algorithms on Piglet dataset in abdomen window.



(a) Low-Dose (b) Normal-Dose (c) BM3D

(d) CNN200 [30] (e) Zhang [58] (f) DRL[3]

(g) DRL-E-M (h) DRL-E-P (i) DRL-E-MP

Figure 6.4: Denoising results of the different algorithms on Piglet dataset in abdomen window.

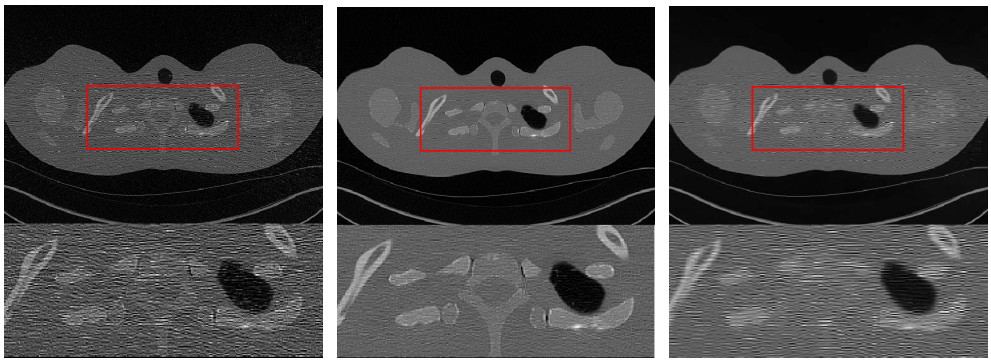
Table 6.3: The average PSNR and SSIM of the different algorithms for the Thoracic dataset.

Metric	Low-dose image	BM3D	CNN200 [30]	Zhang [58]
PSNR	25.66	30.86	33.57	33.73
SSIM	0.4485	0.6552	0.8001	0.8018

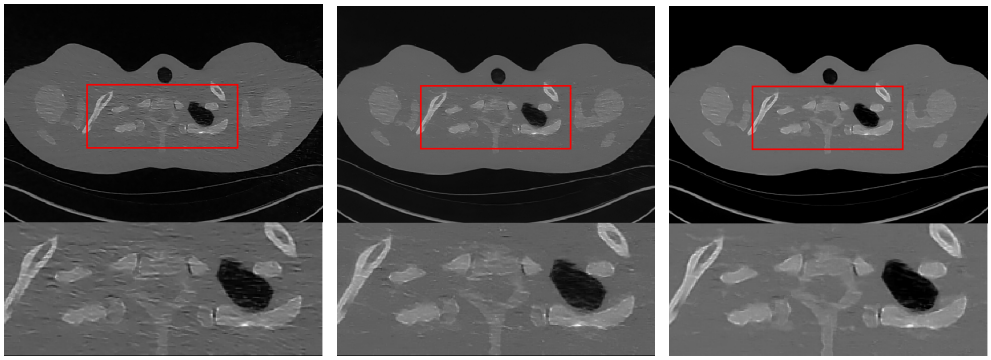
Metric	DRL [3]	DRL-E-M	DRL-E-P	DRL-E-MP
PSNR	34.02	34.03	26.25	31.50
SSIM	0.8059	0.8049	0.4224	0.6381

the edge detection layer.

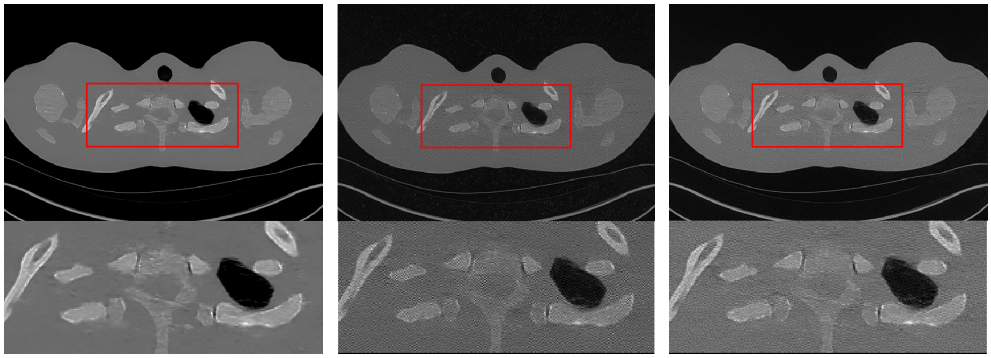
Overall, tables 6.1, 6.2 and 6.3 demonstrate that adding symmetric skip connections and the edge detection layer are powerful tools to enhance the performance of the network. This improvement can be verified by comparing the quantity metrics PSNR and SSIM and also, from the visual comparisons. Figures 6.1to 6.6 confirm that the proposed objective function yeilds more visually appealling results than MSE and perceptual loss.



(a) Low-Dose (b) Normal-Dose (c) BM3D

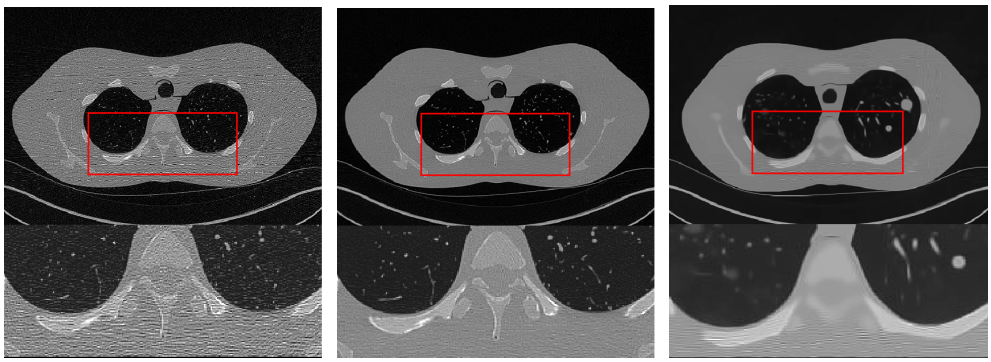


(d) CNN200 [30] (e) Zhang [58] (f) DRL[3]

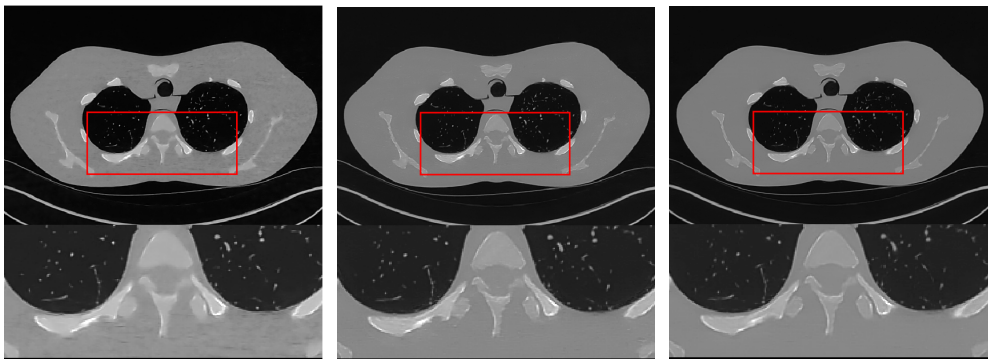


(g) DRL-E-M (h) DRL-E-P (i) DRL-E-MP

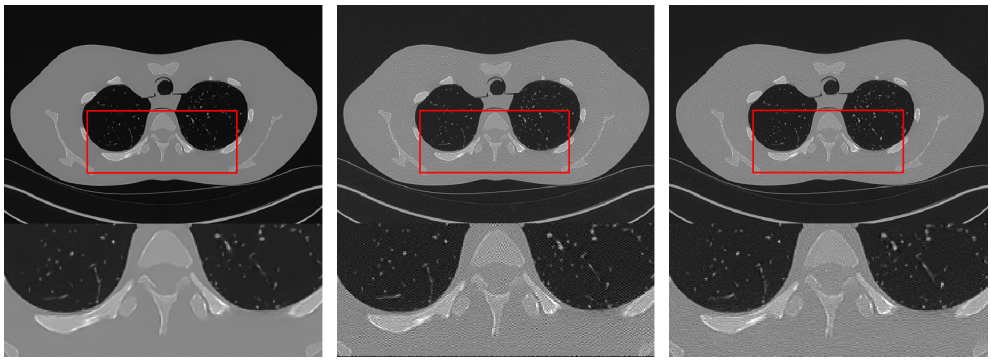
Figure 6.5: Denoising results of the different algorithms on Thoracic dataset in abdomen window.



(a) Low-Dose (b) Normal-Dose (c) BM3D



(d) CNN200 [30] (e) Zhang [58] (f) DRL[3]



(g) DRL-E-M (h) DRL-E-P (i) DRL-E-MP

Figure 6.6: Denoising results of the different algorithms on Thoracic dataset in abdomen window.

Chapter 7

Conclusion and Future Work

Computed tomography is a noninvasive method to see inside the body. In this procedure, X-ray beams are emitted at a patient; and then the beams are collected after passing through the body to generate cross-sectional images. However, studies have shown that exposure to the radiation may lead to diseases such as cancer or tumors, later in life. Low-dose CT imaging helps to reduce the risks of radiations; nevertheless, the reconstructed image is considerably noisy and degraded which affects the confidence of diagnosis.

7.1 Deep Learning for Noise Removal

In this study, a deep neural network is proposed to remove noise from low-dose CT images. The network consists of eight convolutional layers. Dilated convolution, batch normalization, and residual learning are adopted in this network to improve the quality of low-dose CT image. Dilate convolution is used instead of standard convolution, in the middle layers. It helps to increase the receptive field and capture more contextual information in fewer layers. Batch normalization technique facilitates the training process by limiting the gradients from vanishing/exploding in backpropagation phase. It allows using a higher learning rate which accelerates the training. Residual learning also assists in optimizing the network by passing details among non-consecutive layers. This study introduces an edge detection layer that to extract the edge maps of the input image. The layer uses Sobel kernels and does not have any trainable parameters.

The experiments demonstrate that dilated residual learning network outperforms the

state of the art BM3D, CNN200 and [58] networks, when the networks are optimized by mean-square error. Adding the edge detection layer increases the quantity metrics, PSNR and SSIM, even more. The visual comparisons confirm this improvement. However, the experiments suggested that the mean-square error not be the most suitable objective function for CT images. It blurs some of the textural details in the output image. Optimizing by perceptual loss helps to overcome this problem, but it also injects grid-like artifacts to the output. To conquer these problems, a new objective function is defined which is the combination of mean-square error and perceptual loss. This objective function brings out the benefits of each loss while eliminating the mentioned problems.

Three datasets are used to examine the effectiveness of the proposed network, the real piglet dataset, Thoracic dataset, and simulated lung dataset. The latter dataset is generated by adding Poisson noise to the projection of normal-dose lung CT images.

7.2 Future Work

Deep learning is a growing field of artificial intelligence that has been successfully applied to many areas in science. Besides many advancements, deep learning has introduced new loss functions such as perceptual loss and textural loss. It is suggested to examine optimizing a network with these functions alone and in combination with other ones such as mean square error, structural dissimilarity index and adversarial loss. It may lead to finding the most suitable objective function for CT images.

Another idea is to use consecutive CT slices to train a network. In a CT dataset, following CT images are acquired with a little movement of the table/scanner. Therefore, these images are very similar together and share many details. Looking at the consecutive slices usually helps radiologists to understand the details better and detect abnormalities. It is suggested to use such an approach for training a neural network.

References

[1] S. P. Power, F. Moloney, M. Twomey, K. James, O. J. O'Connor, and M. M. Maher, "Computed tomography and patient risk: facts, perceptions and uncertainties," *World journal of radiology*, vol. 8, no. 12, p. 902, 2016.

[2] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on image processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[3] M. Gholizadeh-Ansari, J. Alirezaie, and P. Babyn, "Low-dose CT denoising with dilated residual network," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2018, pp. 5117–5120.

[4] A. C. Kak and M. Slaney, *Principles of computerized tomographic imaging*. IEEE press New York, 1988.

[5] L. Innolitics. (2018) Dicom standard browser @ONLINE. [Online]. Available: https://dicom.innolitics.com/ciods[Accessed7Dec.2018]

[6] G. L. Zeng, "Revisit of the ramp filter," *IEEE Transactions on Nuclear Science*, vol. 62, pp. 131–136, 2015.

[7] M. Beister, D. Kolditz, and W. A. Kalender, "Iterative reconstruction methods in x-ray ct." *Physica medica : PM : an international journal devoted to the applications of physics to medicine and biology : official journal of the Italian Association of Biomedical Physics*, vol. 28 2, pp. 94–108, 2012.

[8] World Health Organisation, "Radiation dose in x-ray and ct exams," 2017, https://www.radiologyinfo.org/en/pdf/safety-xray.pdf, Last accessed on 2018-12-17.

[9] M. Donya, M. Radford, A. ElGuindy, D. Firmin, and M. H. Yacoub, "Radiation in medicine: Origins, risks and aspirations," *Global Cardiology Science and Practice*, p. 57, 2015.

[10] S. P. Raman, M. Mahesh, R. V. Blasko, and E. K. Fishman, "CT scan parameters and radiation dose: practical advice for radiologists." *Journal of the American College of Radiology : JACR*, vol. 10 11, pp. 840–6, 2013.

[11] A. B. Somigliana, G. Zonca, G. Loi, and A. E. Sichirollo, "How thick should CT/MR slices be to plan conformal radiotherapy? a study on the accuracy of three-dimensional volume reconstruction." *Tumori*, vol. 82 5, pp. 470–2, 1996.

[12] M. Alshipli and N. A. Kabir, "Effect of slice thickness on image noise and diagnostic content of single-source-dual energy computed tomography," in *Journal of Physics: Conference Series*, vol. 851, no. 1. IOP Publishing, 2017, p. 012005.

[13] G. Zhang, N. Marshall, R. Jacobs, Q. Liu, and H. Bosmans, "Bowtie filtration for dedicated cone beamCTof the head and neck: a simulation study," *The British journal of radiology*, vol. 86, no. 1028, p. 20130002, 2013.

[14] E. C. Ehman, L. Yu, A. Manduca, A. K. Hara, M. M. Shiung, D. Jondal, D. S. Lake, R. G. Paden, D. J. Blezek, M. R. Bruesewitz *et al.*, "Methods for clinical evaluation of noise reduction techniques in abdominopelvic CT," *Radiographics*, vol. 34, no. 4, pp. 849–862, 2014.

[15] A. Manduca, L. Yu, J. D. Trzasko, N. Khaylova, J. M. Kofler, C. M. McCollough, and J. G. Fletcher, "Projection space denoising with bilateral filtering and CT noise modeling for dose reduction in CT," *Medical physics*, vol. 36, no. 11, pp. 4911–4919, 2009.

[16] P. J. Pickhardt, M. G. Lubner, D. H. Kim, J. Tang, J. A. Ruma, A. M. del Rio, and G.-H. Chen, "Abdominal CT with model-based iterative reconstruction (mbir): initial results of a prospective trial comparing ultralow-dose with standard-dose imaging," *American journal of roentgenology*, vol. 199, no. 6, pp. 1266–1274, 2012.

[17] J. G. Fletcher, K. L. Grant, J. L. Fidler, M. Shiung, L. Yu, J. Wang, B. Schmidt, T. Allmendinger, and C. H. McCollough, "Validation of dual-source single-tube

reconstruction as a method to obtain half-dose images to evaluate radiation dose and noise reduction: phantom and human assessment using CT colonography and sinogram-affirmed iterative reconstruction (safire)," *Journal of computer assisted tomography*, vol. 36, no. 5, pp. 560–569, 2012.

[18] M. K. Kalra, M. Woisetschläger, N. Dahlström, S. Singh, S. Digumarthy, S. Do, H. Pien, P. Quick, B. Schmidt, M. Sedlmair *et al.*, "Sinogram-affirmed iterative reconstruction of low-dose chest CT: effect on image quality and radiation dose," *American Journal of Roentgenology*, vol. 201, no. 2, pp. W235–W244, 2013.

[19] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Processing*, vol. 41, pp. 3397–3415, 1993.

[20] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on.* IEEE, 1993, pp. 40–44.

[21] K. Engan, S. O. Aase, and J. H. Husoy, "Method of optimal directions for frame design," in *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, vol. 5. IEEE, 1999, pp. 2443–2446.

[22] M. Aharon, M. Elad, A. Bruckstein *et al.*, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on signal processing*, vol. 54, no. 11, p. 4311, 2006.

[23] Y. Chen, X. Yin, L. Shi, H. Shu, L. Luo, J.-L. Coatrieux, and C. Toumoulin, "Improving abdomen tumor low-dose CT images using a fast dictionary learning based processing," *Physics in Medicine & Biology*, vol. 58, no. 16, p. 5803, 2013.

[24] K. Abhari, M. Marsousi, J. Alirezaie, and P. Babyn, "Computed tomography image denoising utilizing an efficient sparse coding algorithm," *2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)*, pp. 259–263, 2012.

[25] S. Hashemi, N. S. Paul, S. Beheshti, and R. S. Cobbold, "Adaptively tuned iterative low dose CT image denoising," *Computational and mathematical methods in medicine*, vol. 2015, 2015.

[26] D. Kang, P. Slomka, R. Nakazato, J. Woo, D. S. Berman, C.-C. J. Kuo, and D. Dey, "Image denoising of low-radiation dose coronary CT angiography by an adaptive block-matching 3d algorithm," in *Medical Imaging 2013: Image Processing*, vol. 8669. International Society for Optics and Photonics, 2013, p. 86692G.

[27] S. Xiaobao, D. Yong, D. Ruizhe, and N. Tianye, "A three-dimensional denoising method for low-dose computed tomography," *Journal of Medical Imaging and Health Informatics*, vol. 7, no. 1, pp. 283–287, 2017.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[29] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.

[30] H. Chen, Y. Zhang, W. Zhang, P. Liao, K. Li, J. Zhou, and G. Wang, "Low-dose CT via convolutional neural network," *Biomedical optics express*, vol. 8, no. 2, pp. 679–694, 2017.

[31] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[32] H. Chen, Y. Zhang, M. K. Kalra, F. Lin, Y. Chen, P. Liao, J. Zhou, and G. Wang, "Low-dose CT with a residual encoder-decoder convolutional neural network," *IEEE transactions on medical imaging*, vol. 36, no. 12, pp. 2524–2535, 2017.

[33] M. Nishio, C. Nagashima, S. Hirabayashi, A. Ohnishi, K. Sasaki, T. Sagawa, M. Hamada, and T. Yamashita, "Convolutional auto-encoder for image denoising of ultra-low-dose CT," *Heliyon*, vol. 3, no. 8, p. e00393, 2017.

[34] E. Kang, J. Min, and J. C. Ye, "A deep convolutional neural network using directional wavelets for low-dose x-ray CT reconstruction," *Medical physics*, vol. 44, no. 10, 2017.

[35] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks. arxiv e-prints (june 2014)," *arXiv preprint stat.ML/1406.2661*, 2014.

[36] X. Yi and P. Babyn, "Sharpness-aware low-dose CT denoising using conditional generative adversarial network," *Journal of digital imaging*, pp. 1–15, 2018.

[37] J. M. Wolterink, T. Leiner, M. A. Viergever, and I. Išgum, "Generative adversarial networks for noise reduction in low-dose CT," *IEEE transactions on medical imaging*, vol. 36, no. 12, pp. 2536–2545, 2017.

[38] Q. Yang, P. Yan, Y. Zhang, H. Yu, Y. Shi, X. Mou, M. K. Kalra, Y. Zhang, L. Sun, and G. Wang, "Low dose CT image denoising using a generative adversarial network with wasserstein distance and perceptual loss," *IEEE transactions on medical imaging*, 2018.

[39] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," *arXiv preprint arXiv:1609.04836*, 2016.

[40] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural networks*, vol. 12, no. 1, pp. 145–151, 1999.

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

[42] (2017, dec) Convolutional neural networks. [Online]. Available: http://cs231n.github.io/convolutional-networks/

[43] G. Gwardys. (2017, dec) Convolutional neural networks backpropagation: from intuition to derivation. [Online]. Available: https://grzegorzgwardys.wordpress.com-/2016/04/22/8/

[44] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.

[45] A. Ng, "Sparse autoencoder," *CS294A Lecture notes*, vol. 72, no. 2011, pp. 1–19, 2011.

[46] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

[47] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, 2017.

[48] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2862–2869.

[49] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 479–486.

[50] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Advances in Neural Information Processing Systems*, 2016, pp. 2802–2810.

[51] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[52] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *Advances in neural information processing systems*, 2015, pp. 2017–2025.

[53] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1520–1528.

[54] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[55] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *CoRR*, vol. abs/1511.07122, 2015.

[56] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[57] T. Wang, M. Sun, and K. Hu, "Dilated deep residual network for image denoising," in *Tools with Artificial Intelligence (ICTAI), 2017 IEEE 29th International Conference on*. IEEE, 2017, pp. 1272–1279.

[58] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep cnn denoiser prior for image restoration," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2017.

[59] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*. Springer, 2016, pp. 694–711.

[60] M. S. Sajjadi, B. Schölkopf, and M. Hirsch, "Enhancenet: Single image super-resolution through automated texture synthesis," in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4501–4510.

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

[62] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[63] T. M. Buzug, *Computed tomography: from photon statistics to modern cone-beam CT*. Springer Science & Business Media, 2008.

[64] G. M. McDermott, F. U. Chowdhury, and A. F. Scarsbrook, "Evaluation of noise equivalent count parameters as indicators of adult whole-body fdg-pet image quality," *Annals of nuclear medicine*, vol. 27, no. 9, pp. 855–861, 2013.

[65] T. Chang, G. Chang, J. W. Clark, R. H. Diab, E. M. Rohren, and O. R. Mawlawi, "Reliability of predicting image signal-to-noise ratio using noise equivalent count rate in pet imaging." *Medical physics*, vol. 39 10, pp. 5891–900, 2012.

[66] J. Wang, H. Lu, Z. Liang, D. Eremina, G. Zhang, S. Wang, J. Chen, and J. Manzione, "An experimental study on the noise properties of x-ray CT sinogram data in radon space," *Physics in Medicine & Biology*, vol. 53, no. 12, p. 3327, 2008.

[67] A. Macovski, *Medical imaging systems*. Prentice-Hall Englewood Cliffs, NJ, 1983, vol. 20.

[68] D. Zeng, J. Huang, Z. Bian, S. Niu, H. Zhang, Q. Feng, Z. Liang, and J. Ma, "A simple low-dose x-ray CT simulation from high-dose scan," *IEEE transactions on nuclear science*, vol. 62, no. 5, pp. 2226–2233, 2015.

[69] G. Zuidhof, "Full preprocessing tutorial," February 2017. [Online]. Available: https://www.kaggle.com/gzuidhof/full-preprocessing-tutorial/

[70] T. Szczykutowciz, "Add noise to CT image," September 2012. [Online]. Available: http://www.quarkquark.com/work/Add_Noise_to_CT_Image_ver2.html

[71] W. Lingle, B. Erickson, M. Zuley, R. Jarosz, E. Bonaccio, J. Filippini, and N. Gruszauskas, "Radiology data from the cancer genome atlas breast invasive carcinoma [tcga-brca] collection," *The Cancer Imaging Archive*, 2016.

[72] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle *et al.*, "The cancer imaging archive (tcia): maintaining and operating a public information repository," *Journal of digital imaging*, vol. 26, no. 6, pp. 1045–1057, 2013.

[73] M. A. Gavrielides, L. M. Kinnard, K. J. Myers, J. Peregoy, W. F. Pritchard, R. Zeng, J. Esparza, J. Karanian, and N. Petrick, "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. 15 244–15 255, Jul 2010. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-18-14-15244

[74] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.