APPROVAL SHEET

Title of Dissertation: CT-scan image denoising with Generative Adversarial Networks

Name of Candidate: Binit Gajera

Master of Science, 2020

Dissertation and Abstract Approved: David R. Chapman

Dr. David Chapman

Assistant Professor

Department of Computer Science

and Electrical Engineering

Date Approved: 06/30/2020

ABSTRACT

Title of Thesis: CT-scan image denoising with Generative Adversarial Networks

Binit Gajera, Master of Science, 2020

Thesis directed by: Dr. David Chapman
Department of Computer Science
and Electrical Engineering

We propose a Generative Adversarial Network (GAN) optimized for noise reduction in CT-scans. The objective of CT scan denoising is to obtain higher quality imagery using a lower radiation exposure to the patient. Recent work in computer vision has shown that the use of Charbonnier distance as a term in the perceptual loss of a GAN can improve the performance of image reconstruction and video super resolution. However, the use of a Charbonnier perceptual distance term has not yet been applied or evaluated for the purpose of CT scan denoising. Our proposed GAN makes use of the Wasserstein distance as an adversarial loss function and our perceptual loss combines Charbonnier distance with pre-trained VGG-19. We evaluate our approach using both simulated Poisson noise, as well as real low-dose CT imagery. Our evaluation on real Low-Dose CT (LDCT) imagery applies published methods for estimating the noise through a uniform medium of Air and/or Soft tissue. We evaluate our CT-denoising GAN by measuring the noise reduction over simulated as well as real Low-Dose CT imagery. Our findings show

that the incorporation of the Charbonnier Loss with the VGG-19 network improves the performance of the denoising as measured with Peak Signal-to-noise ratio (PSNR), Structural Similarity Index (SSIM), as well as Air and Soft Tissue noise metrics.

CT-scan image denoising with Generative Adversarial Networks

by

Binit Gajera

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, Baltimore County in partial fulfillment
of the requirements for the degree of
Master of Science
2020

Advisory Committee:
Dr. David Chapman, Chair/Advisor
Dr. Hamed Pirsiavash
Dr. Tim Oates

Dedication

Dedicated towards my parents and my sister for their constant support and unconditional love

Acknowledgments

First and foremost, I would like to thank my advisor/mentor Dr. David Chapman for giving me the opportunity to work on such exciting research and collaborate with other members from the team. I thank Dr. Chapman from the bottom of my heart for providing me with his guidance, support, expertise and uninterrupted time. I thank Dr. Phuong Nguyen as well for providing her guidance whenever I needed it and for giving me the chance to work with her team and gain necessary insights into different research projects. Being a member of the VIPAR research lab has given me a lot of insights into research and has extended my knowledge in various fields.

I thank all my colleagues from the research lab who provided their support and guidance whenever needed and I was glad to get a chance to work amongst such highly intellectual individuals. I also would like to thank Dorsa Ziaie for providing her expertise on all different kinds of matter/situations that I faced, including this thesis, and giving me the support anytime I needed. Apart from that, I would like to thank all my friends from University of Maryland because of whom I would be cherishing my complete graduate experience and to all those friends who provided their continuous support throughout my journey of past two years.

I also thank my committee members Dr. Tim Oates and Dr. Hamed Pirsiavash for giving me their feedback and providing the necessary guidance during my gradute studies.

Lastly, I thank my parents, my sister and all my friends and relatives for giving

me various moments to cherish forever and for providing with their support and faith that lead to the completion of my thesis.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

CT-scan	Computed Tomography scan
LDCT	Low-Dose CT-Scan
NDCT	Normal-Dose CT-Scan
CNN	Convolutional Neural Network
GAN	Generative Adversarial Network
MSE	Mean-Squared Error
PSNR	Peak Signal to Noise Ratio
SSIM	Structural Similarity Index
VGG19	Visual Geometry Group with 19 deep convolutional layers
WGAN	Wasserstein Generative Adversarial Network
CTDI	CT dose index
DLP	Dose-Length product
IQA	Image Quality Assessment
ROI	Region of interest
HU	Hounsfield Units

Chapter 1: Introduction

1.1 Computed Tomography Scan

Computed Tomography (CT) is an x-ray imaging procedure where a narrow beam of x-rays enters the patient's body from all directions at some point during the scan, this creates cross-sectional images or "slices" of the body. Then these slices are in turn used for diagnosis purposes. The scans capture detailed images of internal organs, bones, soft tissue and blood vessels. CT has an advantage of the x-ray modality in that the 3D volume can allow the radiologist to look around bones and other anatomical structures that might disrupt view of important regions thereby hindering diagnosis. However, a disadvantage of CT scanning is that the requirement of taking slices from so many different angles increases the overall amount of radiation exposure. As such, techniques to improve the quality of the CT scan while simultaneously decreasing the radiation exposure of the patient are an important area of research.

The process of performing a CT-scan exposes the patient's body to radiation because of the x-rays entering the body from multiple directions. Here, the dose of radiation can be controlled but that in turn reflects on the quality of the scan as well. The strength of a CT is its ability of visualizing structures of low contrast in

a subject, but that is again dependent on the level of radiation dose used on the subject. It has been noted that, higher the dose contributing to the scan, the less image-noise is present which in turn makes it easy to perceive the low-contrast structures [1]. CT dosimetry is an approach of measuring the amount of radiation dose used for scanning the subject and the dose levels can be controlled by the operator before starting the process of CT-scan. While performing a CT-scan, each slice of tissue receives radiation not only when that slice is scanned but also when the adjacent slices are scanned and thus the patient is exposed to the radiation from all sides and even to the depth of the structure which is being scanned, therefore right amount of radiation dose is important so that the patient does not suffer from any side effects from the scan in the nearby future.

1.2 Radiation Dose

CT dose index (CTDI) is the most commonly used dose descriptor, which represents the dose to a location in a scanned volume. There are various versions of the descriptor such as $CTDI_{100}$ which takes a linear measure of dose distribution over a pencil ionization chamber and hence does not take into account the topographical variation of a human body, $CTDI_w$ is a weighted dose index for periphery and center this makes it more relatable to the human body structure and is used is CT-scan instruments, also $CTDI_{vol}$ is a type of dose index which performs similar to the weighted version but divides $CTDI_w$ by a pitch factor. Any form of CTDI is just an estimate of average radiation dose, there still exists risk from ionizing

radiation which is more closely related to the total amount of the radiation dose deposited in the patient. A more close estimate of the dose index would be suggested Dose-length product (DLP):

$$DLP = L \times CTDI_{vol} \tag{1.1}$$

Here L is the total z-direction length of the examination which gives it the relation with the level of depth as well. Some CT scanners also display DLP alongside the CTDI for operators.

1.3 Visualizing CT-scans

Fundamentally, image quality in CT-scans depends on 4 basic factors as applied to all medical imaging, image contrast, spatial resolution, image noise and artifacts. We here would be discussing image noise, as our model and data correlates with that information. As per [1], the graininess in a Low-Dose CT-scan is of the same nature as radiographic quantum mottle which is caused due to the use of a limited number of photons to form the image. The distribution that we obtain from the quantum mottle is fundamentally similar to the Poisson Distribution which we are adding to our dataset while preparing the data for the model. Now, the main task for a CT-scan is to visualize low-contrast structure which in turn is primarily dependent on image noise in such manner that if the noise is more in the image the structures would not be visible properly to the viewer of the scan and hence less amount of noise is preferred to get the correct representation and

state of the low-contrast structure.

Visualizing low-contrast structures with high quality in turn means exposing the patient high amounts of radiation because a Low-Dose CT-scan (LDCT) would contain noise grains in the slices and thus that makes it difficult to perceive the structures. Our task here is to achieve a better visualization of the low-contrast structures in a LDCT scan by decreasing the amount of noise present in those slices, the LDCT scan should in turn be at least visually similar to the NDCT scan so that patients can avoid getting exposed to high radiation dose and medical doctors can also diagnose, without any trouble and extra analysis, using the LDCT scan.

1.4 Using GANs

Therefore, to reduce the amount of radiation a patient would be facing while performing their regular CT-scans, we propose a CT denoising network that can denoise scans from LDCT so that they are more perceptive and more close to NDCT scans. This would help the radiologists as well because many at time they have to perform LDCT scans on patients due to number of reasons, and in doing so it would help and be much better if the scans had lower amount of grains in them so that the low-contrast structures, the soft tissues and the artifacts can be visually perceived without any constraints or more analysis. Also, the CT denoising would be considered as a postprocessing algorithm after the actual scan has been performed, this is because once the scan is extracted from lower radiation

dose it would have some amount of grains/noise in it which our goal is to reduce, hence after the scan and before the radiologist takes the scan we would have to perform a post processing step of decreasing the number of grains and decreasing the level of noise from the slices, where our denoising network would be used. Generative Adversarial Networks are used for many applications such as Art creation [2], Image super-resolution [3], and Image transformation [4]. Our end goal here is to generate an image with a lower amount of noise then before, and hence we decided that we can incorporate the network of GAN and use it to achieve our functional goal. WGAN is preferred here over GAN because most of the task with GAN does not require the generated images to be in the same distribution as the training dataset while for us and other tasks such as Image super-resolution it is imperative that the generated image be in the same distribution as the training dataset and hence we decided to use WGAN. Also, there are issues with training a normal GAN such as vanishing gradient because of which the generator of the GAN does not converge and takes more amount of time to train the model, this disadvantage of GAN is removed from WGAN by using a different loss function whose backpropagation is possible at all unit of time during training, this function is the Earth-mover or Wasserstein distance. We would be discussing more about the GANs and the Problems with GANs in the following sections.

1.5 Loss functions and its integration with WGAN

To maintain the feature space between the generated distribution and the training set distribution we have incorporated the use of Perceptual Loss [3] into our WGAN. The loss function which we would also discuss in the following sections contains a summation function of the perceptual loss and the Wasserstein distance function. The main usage behind the perceptual loss is to maintain the human perceptive features in the images. Several times just using GANs loss function can add artifacts or blurring which most of the times have been added because of the MSE loss function required by those tasks [3, 5], since our task relates to medical imaging where it is very crucial to maintain the features we use perceptual loss instead of the MSE loss. The perceptual loss in turn does require extracted features from the slices, for which we are also using a pre-trained VGG 19 model as a feature extractor. The VGG-19 [6] is a convolutional neural network with 19 deep layers which is trained on ImageNet that contains 14 million images, given the size of the dataset it would not be physically possible to train the model again on the same dataset and generate the weights here for our usage instead it is more beneficial to use the pre-trained model provided by the authors, this model would have the same weights and would be able to extract necessary features from the CT slices because it is trained on a real world dataset of ImageNet [7].

Over the past few years, many researchers have been working on the Iterative Algorithms (IR) for LDCT image reconstruction. It has been shown that LDCT image reconstruction in turn helps in decreasing the amount of noise as well [8] but

they still may lose some details. Apart from that the bottleneck for these applications to be used practically is that they have a high amount of computational cost and thus necessary resources to perform and execute the given algorithms are needed. Wolterink et. al [9] were the first to apply GANs for the purpose of CT-scan denoising and it did show promising results about which we will discuss more in the Related Work section of this paper.

1.6 Evaluation of the CT-scans

To evaluate the model we make use of the PSNR ratio between the generated and the NDCT slices during training which is compared with the PSNR ratio between the same NDCT slice and its corresponding LDCT slice. After the training is completed and testing slices are generated we also compare the MSE Loss between those slices to see improvement and decrement of noise levels. Apart from these standard metrics, we also make use of published evaluation method [10] for CT-scans which makes use of Standard Deviation and Variance to find the noise levels in the region of soft tissues and air pixels after the pixel values have been converted to Hounsfield units in the preprocessing step. Also, the method uses Sobel filter to detect the region of interest (ROI) in the slices and further modification have been done to find the ROIs for our data distribution. More about the published approach and the results obtained would be discussed in the later part of the paper.

1.7 Thesis Statement

A Generative Adversarial Network that incorporates Charbonnier distance along with the perceptual loss and adversarial loss can improve the quality of CT scans without increasing the radiation exposure to the patient.

1.8 Contributions

- We introduce the Charbonnier distance term as part of the perceptual loss of a CT denoising GAN. Although the Charbonnier loss term has been recently demonstrated for related video-superresolution, to the best of our knowledge we are the first to evaluate this approach for CT-scan denoising.
- We compare the results of our CT-scan denoising algorithm against two state-of-the-art CT-denoising GANs from literature, and demonstrate that the proposed CL-WGAN outperforms these techniques in terms of PSNR and SSIM.
- We evaluate the performance of the CL-WGAN using published soft-tissue and smooth region noise metrics from Radiology literature in addition to PSNR and SSIM.

1.9 Organization of Thesis

The rest of the thesis is organized as follows. We discuss some related works around denoising CT-scan with different approaches and how complex networks have been

used in Chapter 2. Following that we would be discussing more in depth about the components used in building the network and some drawbacks of the Generative Adversarial Networks in Chapter 3. After that we take a look at how the task of denoising is achieved with specific methodology and the datasets used in Chapter 4. We discuss few results visually and in terms of quantitative metrics as well along with a comparison of the approach with similar other approaches in Chapter 5.2. We finally end the thesis with Conclusion in Chapter 6 and discuss some possible future works in Chapter 7.

Chapter 2: Related Work

As mentioned before, there are many algorithms that make use of GANs for image quality assessment and for reducing noise levels in natural image datasets, we would be discussing some of those models and algorithms here. Amongst the approaches that make use of Deep Learning, there are many Iterative Algorithms too that researchers have researched upon for image reconstruction and it has been shown that image reconstruction helps in improving the quality of image in Low-Dose CT-scans which also will be discussed here. Other than using the approaches from deep learning there are mainly two approaches that focus on LDCT denoising that is

- Sinogram filtration
- Iterative algorithms

We would be starting our discussion on related works with these algorithms and later on discuss some of the approaches that utilize the methodology of GANs.

2.1 Sinogram filtration

For CT-scans, a sinogram is nothing but raw data that contains the 2-D array of the projections. The plane representation angular parameter and distance along the projection direction is what the sinogram is. Mainly sinograms are used for image reconstruction, but many researchers such as Davood Karimi et. al [11] were able to design an algorithm that performs filtration using the sinogram data after the image reconstruction has been performed. This in turn provides a CT-scan with lower noise than the original.

One other similar approach to the one mentioned above is performed by Armando Manduca et al. [12] which uses sinograms. They also perform filtration but here the authors have designed an algorithm that uses Sinogram smoothing with bilateral filtering. The authors have proposed the idea of using bilateral filtering directly after the image reconstruction and according to their analysis sharper edges are well suited to techniques like bilateral filtering but the noise model in image space is very complex and hence they apply bilateral filtering on projection space which in turn also produces a denoised scan from Low-Dose CT-scan.

2.2 Iterative algorithms

Iterative algorithms are mainly used for image construction using different techniques such as a statistical noise model [13–15] and prior image information such as sinogram in the image domain. Some other image priors are Total Variation

[16] and dictionary learning [17]. These algorithms can be used for reconstructing images and then apply different filtration methods to reconstruct an image with lower amount of noise than the original CT-scan.

These algorithms have provided convincing results so far, but the only disadvantage of using them is that the resources required to run the algorithms are too many and scarcely available. For example the image priors for iterative algorithms and the sinogram raw data for sinogram filtration is not readily available as the CT-scan itself. Also, these approaches require a high amount of computational cost as mentioned before and thus that also can be considered as a disadvantage of using such methods.

2.3 Image quality assessment and LDCT Denoising using GANs

There are many different approaches that have been researched upon by many researchers for improving the quality of an image using GANs and performing Image Quality Assessment (IQA). One such approach is followed by Wei et. al [18] which incorporates the knowledge gained from the IQA metrics into the GAN model because as per the analysis from the authors using only GAN with convolutional neural networks makes it less robust to blur and noise from which noise is of our main concern. The IQA used here returns a score which can then be given to GAN for efficient learning. They have used another network such as VGG-16, Inception-v2, and MobileNet as a classifier which classifies between artifacts, noise and blur based on which they predict a score for the quality of an image.

The NIMA is an image quality assessment network used by the authors which has been incorporated with a classifier as described above.

Wolterink et. al [9] were the first authors to build a GAN for Low-Dose CT-scan denoising. The GAN constructed by them incorporates voxelwise loss. The generator and discriminator for the GAN are both convolutional neural networks. They have evaluated the results of denoising on mainly three training strategies: Only voxelwise loss, voxelwise and adversarial loss and the third used only adversarial loss. It was reported that they were able to achieve the highest PSNR ratio with only voxelwise loss but lost image statistics while the adversarial loss was able to capture the image statistics of routine-dose scans better and thus maintaining the original features necessary in a CT-scan.

One other such approach where authors have incorporated another metric to train the GAN more efficiently is by using a sharpness detection network by Xin Yi et. al [19]. Their GAN is a Sharpness Aware GAN in the sense that for Low-Dose CT scans they were able to use a conditional Generative Adversarial network and train a sharpness detection network along with the GAN to guide the training process of GAN itself. The results obtained by this approach are fairly pretty good and the amount of noise removed from the LDCT scans does perceptively make a difference in making any type of diagnosis for radiologists.

One of the few accomplishments from the field of machine learning is the Resnet architecture developed by Kaiming He et al. [20]. A Resnet follows the concept of residual learning and forming skip connection layers between the convolutional layers to preserve features and a stable back-propogation for deep CNNs, following

the same learning in GANs is made possible by Chenyu You et al. [21], who proposed an architecture for GANs that contain Skip connection layers and network in network model. The generator G proposed by the authors contains a Feature Extraction Network and a Reconstruction Network that focuses on extracting and maintaining the original features from the CT scan so that the generated slices are a close representation of the normal dose CT slices but with suppressed noise levels. The discriminator in the complete model is a vanilla CNN with few convolutional layers and few fully connected layers at the end. The authors use L1 loss function for the GAN and to stabilize the training of the GAN they use Wasserstein GAN with gradient penalty which contains an adversarial loss as well. The authors claim that the results obtained using their network design suggests that the proposed method can be generalized to various medical image denoising problems but further efforts are needed for training, validation, testing and optimization [21].

One of the other approaches for medical image denoising by the group of authors Chenyu You et al. [22], is shown to give suppressed levels of noise in Low-Dose CT scan by using Wasserstein GAN along with a Structural loss. The network of the Generator contains multiple convolutional layers and the network for the Discriminator is similar to the one discussed before, consisting of vanilla CNN layers. The loss function incorporated by the authors along with the adversarial loss is termed as Structural loss and is defined using the equation for the Structural similarity index metric [23]. The authors also mention that the purpose for using the multi-scale SSIM for creating the loss function is to preserve high-resolution and

critical features for diagnosis. The authors were able to show suppressed levels of noise from Low-Dose CT scan by incorporating the appropriate loss function in their WGAN.

In terms of using GANs for generating slices with lower amount of noise from Low-Dose scans, it is important that the generator does not add any artificial features to the scans and generate the scan which maintains all the features required for diagnosis and are present in the normal-dose CT scan as well. Eunhee Kang et al. [24] proposes a network architecture that specifically targets the importance of maintaining the features of the scan. They propose an architecture that has two discriminators and two generators and three loss functions which includes the adversarial loss, cyclic loss and additionally identity loss. Since the GAN network proposed here has multiple components the loss functions defined by authors that cyclic loss and identity loss use the information from each component to build a scan that suppresses noise levels and maintain the feature space at the same time. The feature space is important to maintain and thus even our model incorporates the use of perceptual loss to extract features and maintain them in our generated scans. The authors [24] mentioned above also provide extensive analysis on the scans and claim that their proposed method is good at reducing the noise in the input low-dose CT images while maintaining the texture and edge information.

Chapter 3: Background

3.1 Generative Adversarial Networks (GANs)

There are many machine learning models nowadays that can perform tasks that achieve human-level accuracies, but not all tasks that run using machine learning achieve those results. Generating an image that is never seen before by anyone and still maintain the features that humans perceive in it is an important and notable breakthrough in the field of machine learning and we were able to achieve those results by using a Generative Adversarial Network (GAN) proposed by Ian J. Goodfellow et. al [25]. Our network here uses a variant of the GAN version proposed by the original author and hence it is necessary to discuss the basic implementation of a GAN here.

A Generative Adversarial Network is a network that consists of two other networks from which one works as a generative network and the other works as a discriminative network. The generative model would be termed as G here, and the discriminative would be termed as D. From both networks, the desired result would be obtained from the generative model given that we need something new, it does that by capturing the data distribution provided to the GAN. While the discriminative model would be estimating a probability of a sample coming from

the training data and not from the Generator (G). Both of the models are supposed to train each other and solve the minimax problem that will be described later. During the training, G is supposed to maximize the probability of D making a mistake that is G is supposed to produce results in such a manner that D labels those results coming from the training data rather than from G.

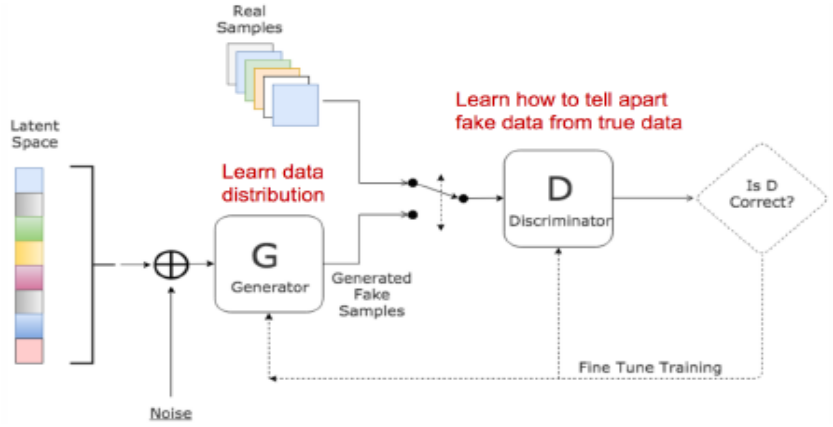


Figure 3.1: Brief overview of the GAN architecure by [26]

In simpler terms, let's assume the generator in our GAN to be a forger of arts which means it's task is to create duplicates of the original art. The discriminator here is assigned the task to verify the authenticity of the art, that is, given an art to verify that it is fake or is it drawn by the original author. The discriminator would not be knowing from which distribution it is getting the arts so the art can be real as well fake. When the training starts G would start with drawing random noise and D would be predicting them as False seeing which G's parameters get updated and it improves upon its results as new and new data it gets to see. The D then should be able to recognize G's fake images in order to increase its own accuracy and thus both of the models would be learning from each other until both

of them start producing expected results.

Therefore we can say that we train D to maximize the probability of assigning the correct label to both training samples as well samples from G. We simultaneously train G to minimize $log(1 - D(G(z)))$ where G(x) is a differentiable function represented by the generator network and D(y) represents the probability of y coming from the training sample and not from G. The minimax game that these both networks play with each other can be termed as $V(D, G)$ and can be formulated as described in equation 3.1.

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z \sim p_z(z)}[log(1 - D(G(z)))] \tag{3.1}$$

Here, $P_z(z)$ represents an input noise variable considered as prior for the generator and $P_{data}(x)$ represents the input data that is the training data provided to GAN.

3.2 Problems in GANs

Although GAN has shown success in many realistic image generation tasks, the training is not that easy, the process is known to be slow and unstable. A GAN without any modifications contains several problems that makes its training very difficult and may not give expected results every time. There are two common divergences used in Generative models Kullback–Leibler (KL) divergence and Jensen–Shannon (JS) divergence.

3.2.1 Kullback–Leibler Divergence

KL Divergence measures how a probability distribution p diverges from a second expected probability distribution q. The minimum divergence here that we can achieve is zero which will be when $p(x) == q(x)$. KL divergence is also asymmetric which means, when $p(x)$ is close to zero but $q(x)$ is significantly greater than zero, then q's effect is disregarded because we measure how p diverges from q [26]. Now, this can give inaccurate results when we just want to measure the similarity between two equally important distributions.

3.2.2 Jensen–Shannon Divergence

JS Divergence is also a measure of similarity between two probability distributions, but it is bounded by [0, 1]. The JS Divergence is symmetric and more smooth than the KL Divergence.

One of the main issues of the GAN discussed above is of Vanishing gradient which makes the training difficult. Suppose if the discriminator D achieves 100% accuracy, that is it becomes perfect, then in that case the Loss would become zero because of which we end up with no gradient to update the loss during the iterations. Therefore, there are mainly two concerns that arise from this,

- If the discriminator performs well, then the gradient of the loss function drops down close to zero and the learning slows down or may stop completely

- If the discriminator does not perform at all, then the generator would not have its appropriate feedback and the loss function would be unable to represent the reality

Because of such reasons it becomes difficult to fully train a Generative Adversarial Network and receive expected or even good results.

3.3 Wasserstein GAN

Considering the instability of GAN, Wasserstein GAN is a variant of the same. The main concern in the GAN is of using the divergence as loss functions for generator and discriminator so we used a completely different metric here for loss function which would increase the performance of the GAN even when discriminator shows great results. The modification and network architecture for Wasserstein GAN proposed by Arjovsky et al. [27], has shown better results than GAN [26] in my experiments including tasks related to medical imaging, thus we would be using WGAN for our further research.

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}[\| x - y \|] \tag{3.2}$$

Earth-Mover (EM) distance or Wasserstein-1 is a type of distance formula that we can use to construct the loss function for the Wasserstein GAN, in fact the name of the GAN also comes from the name of the distance formula. Here, for the distance, we would be using a set of joint distributions whose marginals are $\mathbb{P}_r$

and $\mathbb{P}_g$. Here $\gamma(x, y)$ represent how much "mass" is required to transport from x to y so that the distribution $\mathbb{P}_r$ is transformed into the distribution $\mathbb{P}_g$ [27]. The EM distance then becomes the "cost" of the optimal transport plan.

The authors [27] also discuss about Kantorovich-Rubinstein duality [28] which formulates the equation described in equation 3.3.

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x \sim \mathbb{P}_r}[f(x)] - \mathbb{E}_{x \sim \mathbb{P}_\theta}[f(x)] \tag{3.3}$$

Where supremum is over all the 1-Lipschitz functions f. As per the authors, if we have a parameterized family of functions which in our case would be related to the discriminator and generator, then we could incorporate equation 3.3 into our GAN's loss function such that those family of functions are all K-Lipschitz functions from some K.

The new loss function suggested by the authors [27, 29] that improves the training of GAN for WGAN is formulated in equation 3.4.

$$\min_G \max_D L_{WGAN}(D, G) = -\mathbb{E}_x[D(x)] + \mathbb{E}_z[D(G(z))] + \lambda \mathbb{E}_{\hat{x}}[(\| \bigtriangledown_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \tag{3.4}$$

Here it is another minimax problem that needs to be solved by the network after all the architecture is still of a GAN. The first two terms perform the Wasserstein distance estimation using the EM distance equation with the duality [28], and the last term is added for network regularization which is a gradient penalty

term suggested by I. Gulrajani et al. [29]. $\hat{x}$ is uniformly sampled along straight lines connecting pairs of generated and real samples and λ is a constant weighting parameter [30]. From the equation, we can see that WGAN removes the log function in the losses and also drops the last sigmoid layer in the implementation of the discriminator D.

The discriminator D is now not the direct critic of telling the fake samples apart from real ones but is instead trained to learn a K-Lipschitz continuous function to help compute the Wasserstein distance [26]. To maintain the continuity of the function, the authors [27] suggests clamping the weights to a small window such as $[-0.01, 0.01]$ after every gradient update, this would preserve the Lipschitz continuity. Now as the loss function decreases in training, the Wasserstein distance gets smaller and the generator's output becomes closer to the real data distribution. It is to be noted though, that the loss function is configured to measure the Wasserstein distance between the real distribution $\mathbb{P}_r$ and fake distribution $\mathbb{P}_g$, as required.

As discussed above, clipping the weight in turn again causes instability in long-term training. The WGAN suffers with slow training because of that and is not recommended. To negate the issue of clipping weight the author in collaboration with I. Gulrajani et. al [29] came with the term of gradient penalty which is added in our loss function and which works a regularization parameter to preserve the Lipschitz continuity and also helps us avoid weight clipping. The main ideology behind clipping weight is to have a gradient norm of 1 for f between the points interpolated from real and generated data. Therefore, instead of applying

the clipping, if the gradient norm moves away from its target norm value 1 then we penalize the model using our gradient penalty term.

In GAN, the loss function measures how well it fools the discriminator rather than a measure of the image quality. Therefore in GAN, even when the image quality improves, the generator loss does not drop and it becomes difficult to perform evaluations except visually. On the contrary, the WGAN loss function reflects the image quality as well along with the difference between the real and fake distributions, which is desirable for our experiments.

3.4 Perceptual Loss

Generative Adversarial Networks use generative models that start generating images from noise patterns during training then as the loss decreases or comes closer to zero it produces the images as the same as the real distribution. Here, for our task it is imperative that we maintain the low-contrast structures in the CT-scan and the rest of the information present in the slices. For this to happen we would be using the perceptual loss function along with a VGG-19 network [6].

Many networks with similar needs as ours have used Mean squared error (MSE) loss function for the same reason, here MSE tries to minimize the pixel-wise error between the image patches from both data distributions. However, the MSE loss can generate blurry images or cause distortion or loss of details in the slices [3, 5], which if happened defies the purpose of denoising CT-scans. Therefore, rather than using the MSE loss function it is more advantageous and fruitful to use Per-

ceptual Loss function [3].

$$L_{PL} = \ell^{\phi,j}_{feat}(\hat{y}, y) = \frac{1}{C_j H_j W_j} \|\phi_j(\hat{y}) - \phi_j(y)\|_2^2 \tag{3.5}$$

The equation 3.5 is the perceptual loss function, where C, H, and W represents depth, height and width. $\hat{y}$ represents the result obtained from the generator and y represents the image patch from the real data distribution which for us would be the NDCT data distribution. ϕ_j represents the feature extractor, here VGG-19 would be used as a feature extractor.

The VGG-19 network here works as a feature extractor so that proper comparison can be made between the generator's output and the ground truth distribution in terms of the extracted features. We are using a pre-trained VGG model which takes color images as input, since our CT-images are grayscale we duplicate the CT slices to make RGB channels before feeding to the VGG network. The VGG network itself has 16 convolutional layers and 3 fully connected layers [30]. The output obtained from the last convolutional layer is the feature that is extracted by the VGG network and is used in the perceptual loss function. One of the many reasons for using pretrained VGG-19 model is that it maintains the features of the images, which for a CT-scan is important so that the medical personnels can diagnose correctly. It also provides expected results because it is previously computed on a very large natural image dataset [7].

3.5 Charbonnier Loss

Here, the generator G is tasked at generating slices with suppressed amounts of noise levels from Low-Dose CT scans, one of the possible outcomes in such cases is to generate blurry slices of scans. Now as discussed previously MSE Loss can induce some amount of blur in the images [3, 5] but the same cannot be said for L1 Loss. Here we use a loss function proposed by Charbonnier et al. [31] which can also be used as a regularizer for the GAN.

Using only the adversarial loss of the GAN can construct artifacts in the images and show ringing patterns or unnecessary edges around any object as shown by Alice Lucas et al. [32]. Such patterns can be regularized in GAN by adding a regularizer with the loss function. One of the common regularizers for image synthesis corresponds to the distance estimate between the generated image and the ground truth image available from the training dataset. Hence, we would be using the equation 3.6 proposed by Pierre Charbonnier et al. [31].

$$L_{CL} = C(\hat{x}, x) = \sum_{i} \sum_{j} \sqrt{(\hat{x}_{i,j} - x_{i,j})^2 + \epsilon^2} \tag{3.6}$$

This equation is termed as Charbonnier Loss and is often referred as pseudo-huber loss as well because it resembles the equation of Huber Loss [33]. Other than that it has been shown by Jonathan Brown [34] that the loss function combines the properties of L2 loss and L1 loss by being strongly convex when close to target/minimum and less steep for extreme values [34]. Also according to the

authors the loss function is an adaptive and robust loss function and thus we decided to add to our GAN's loss function. In the equation 3.6, L_{CL} references the Charbonnier Loss that we would be using before, i and j refer to pixel coordinates, $\hat{x}$ represents the estimated image obtained from the last layer of the generator and x represent the ground truth NDCT image. ϵ is a small constant that is preferred to be kept close to zero and thus in our case it is set to $\epsilon = 0.001$. We also found that using the Charbonnier loss instead of MSE Loss in pixel and feature-space provides better results visually and has less amount of blur than that induced by MSE Loss. The features in the slices are maintained by using the Perceptual loss and thus the Charbonnier Loss helps in regularizing the GAN to reduce the artifacts in the generated slices.

Chapter 4: Methodology

4.1 Dataset

For a given machine learning model, using the correct set of data and collecting the right amount of data, is very important. In general terms, a GAN would require a very large amount of data for training the model as there are two neural networks incorporated in it, which train simultaneously as will be seen from the GAN architecture in the paper. Here, for training purposes, we are using the dataset provided by Kaggle Super Bowl 2017 Data Science competition [35]. The dataset contains thousands of high-resolution DICOM lung CT-scans which were originally obtained from the National Cancer Institute. All of the scans are obtained from high-risk patients and further, we preprocess the data to meet our needs so that we can train a model that can identify and mitigate noise from the CT-scans.

Each image in the original data without any preprocessing contains a series with multiple axial slices of the chest cavity and a variable number of 2D slices. Therefore, from each patient, we extract 20 slices where each slice is in DICOM format with the necessary header information available such as slice_thickness, rescale_intercept, rescale_slope, and pixel_array. We used 75 patients out of 1200 for which each pa-

tient has at least 20 slices so that the model can be trained on the provided patients. Out of the 75 patients examinatios, we split the data in train, validation and testing. 65 patients slices have been used for training the Generative Adversarial Network, 6 patient's examinations have been used as validation distribution and 4 patient's examination have been used as test distribution.

4.2 Simulating Low-dose CT-scans

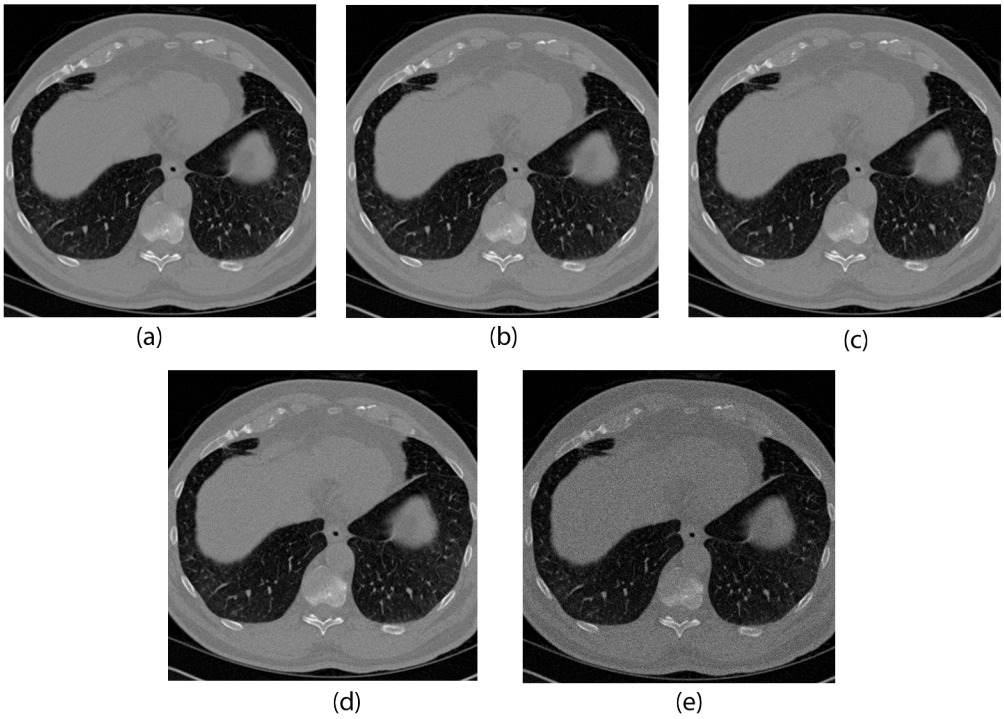


Figure 4.1: Noise classes in the following order (a) GOOD, (b) AVG_GOOD, (c) NEUTRAL, (d) AVG_BAD, (e) BAD

Given that the goal of our machine learning model is to reduce the noise in each CT-scan of the patient, we would need separate data with respect to the original DICOM scans. Therefore, we created visually similar scans to the original

dataset which had some amount of noise in it, here we have created 5 amounts of noise levels as shown in the image. The noise to each slice is added in the form of Poisson distribution. The labels created are BAD, AVG_BAD, NEUTRAL, AVG_GOOD, and GOOD with the respective interval values of adding noise with standard deviation of 10, 50, 120, 180 and 0 Hounsfield units. Here we can see that the dataset labeled as GOOD does not have any type of noise incorporated in it and thus we would be considering those patient slices as Normal-dose CT scans for our training purposes and the dataset labeled as AVG_BAD would be used as low-dose CT scans for training which has noise added to the slices. Moving forward from here, Normal-Dose CT represents the dataset labeled GOOD and Low-Dose CT represents the dataset AVG_BAD.

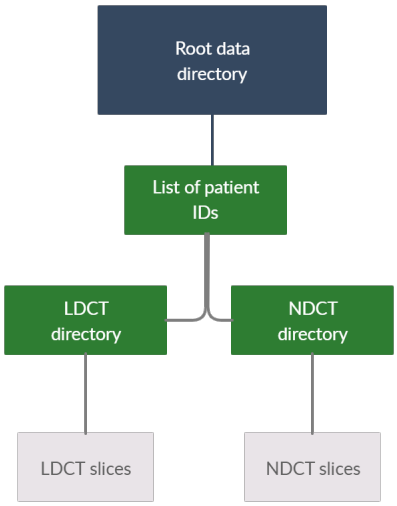


Figure 4.2: Directory tree for training data

To add the Poisson noise to the original dataset it was imperative for us to convert the DICOM slices to NumPy arrays so that appropriate pixel-level modifications can be made to the slices, but once the slices were converted to Numpy for-

mat, it is not possible to convert them into DICOM again as there are certain attributes needed in the DICOM file obtained from the scanner and thus while saving the modified datasets we store few attributes of DICOM such as Slice Thickness, Rescale Intercept and Rescale Slope along with the slice's pixel values. Now, once the datasets have been prepared we modify the directory structure and create a different structure such that the Machine Learning model can read the input data for training as well testing purposes. The directory structure that has been designed is shown in the figure 4.2 where each patient ID will be having 2 subdirectories named LDCT and NDCT.

4.3 Data preprocessing

Since we already simulate the dataset for our respective category the machine learning model would be taking the training data in the format of Numpy arrays which are already read from the Dicom pixel arrays during simulation. Therefore all slices of the scan which are given as an input to the model are in the format of Numpy arrays which stores the Pixel array, Slice Thickness, Rescale Intercept and Rescale Slope obtained from the original metadata of the DICOM scanned file. We also convert the pixel values to Hounsfield Units in order to distinguish between air and the cell structure in the CT-scans, this is considered as an important pre-processing step when dealing with CT-scans. The values such as Slice Thickness, Rescale Intercept and Rescale Slope are needed for the pixel values to be converted to Hounsfiled units and this were stored beforehand in the numpy

arrays during simulation and are being read by the machine learning model accordingly.

Apart from that, we normalize the slices too among the specified range in the model while reading the training data. We also train the model on image patches and thus the patches are created for each scan randomly while reading the CT slices. The patches are formed to increase the dataset so that the GAN can learn appropriate mapping, the training is performed on the patches but the validation data and test data are given to the model as whole images so that our goal of denoising a complete image is preserved. To maintain this patches among the batch data we use Queue, Threading and Tensorflow Coordinators which also provide support to the GAN training by reading the data and maintaining it faster than reading the data normally in a single thread.

4.4 Loss Function

Overall using the equation 3.4 and 3.5 and 3.6 we formulate a combined equation that represents the complete loss function for our WGAN network as shown in equation 4.1.

$$\min_{G}\{\lambda_1[\max_{D} L_{WGAN}(D,G)] + \lambda_2 L_{PL}(G) + (1-\lambda_1-\lambda_2)L_{CL}(G)\} \tag{4.1}$$

Here λ_1 and λ_2 are used as weighting parameters to control the trade-off between the three loss functions that is between WGAN adversarial loss L_{WGAN} and the Perceptual loss from VGG L_{PL} and the Charbonnier Loss L_{CL}. The perceptual

loss here suppresses the noise by comparing the perceptual features of a denoised output generated from the generator against the ground truth NDCT in an established feature space, while the GAN focuses more on migrating the data noise distribution from strong to weak statistically to achieve results that are more close to NDCT distribution. And the Charbonnier loss acts as a regularizer to the GAN training which supports the model by reducing the generation of any artifacts in the generator's output. The weights $\lambda_1 > 0$ and $\lambda_2 > 0$ with $\lambda_1 + \lambda_2 < 1$ are hyper-parameters and are determined experimentally.

4.5 Network Architecture

The overall architecture for the complete WGAN mainly consists of three parts. The parts used are already discussed above but here would see the architecture which defines which part interacts where and the details of the same.

The first part that we have is the generator itself which is a convolutional neural network with 8 convolutional layers, following that we have kept small 3 X 3 kernels in each convolutional layer. The figure 4.3 represents the overall architecture of the model but we can also see the complete network for the Generator, here n represents the number of convolutional kernels, s represents for convolutional stride and k represent the size of the convolutional kernel as a square. For example, *n32s1k3* means 32 convolutional kernels with stride 1 and size 3 X 3. Each of the first seven layers here in the Generator has 32 filters and only the last generates the feature map with a single 3 X 3 filter, which in turn is also the output of

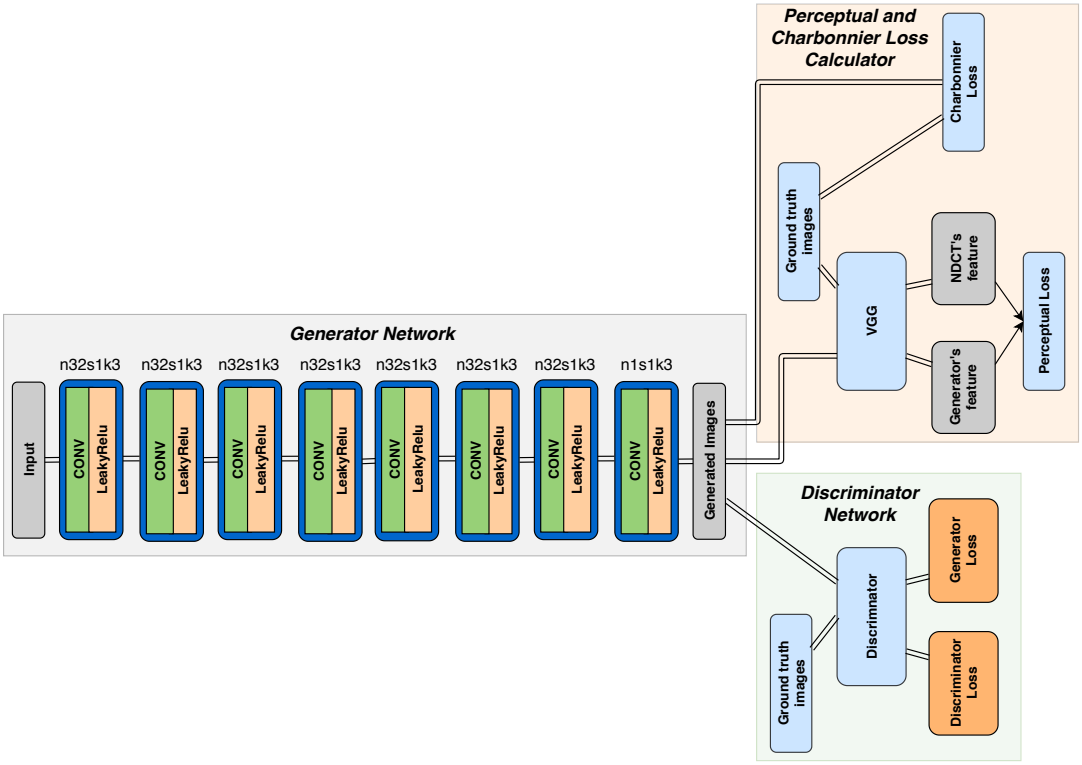


Figure 4.3: Overall architecture of the network

the generator G. Apart from the convolutional layers we also need an activation function after each layer and here we use the Rectified Linear Unit [36].

The second module of the architecture is the pre-trained VGG network which extracts features and returns them to the perceptual loss function which in turn returns the loss value. The VGG would take the denoised output from the generator G as input and would return the extracted features required to calculate perceptual loss. The perceptual loss would then be calculated using previously mentioned Eq. 3.5. As soon as we get the error value we update the weights of only G and keep the weight parameters for the pre-trained network intact as we do not have to update them. The second module also includes the calculation of the Charbonnier loss as shown in the figure, it is calculated alongside the calcula-

tion of the perceptual loss.

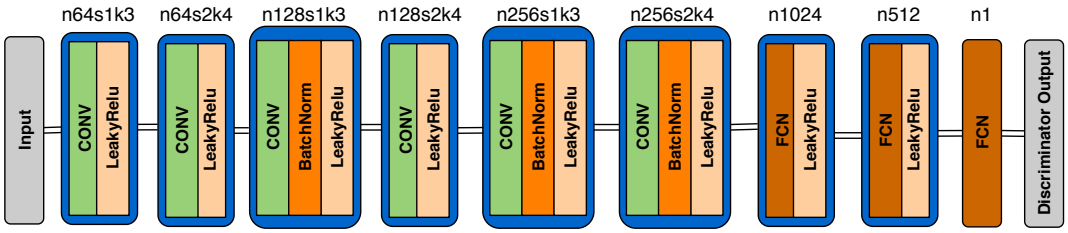


Figure 4.4: Discriminator

The third and the final part of the network architecture is the discriminator D itself which is shown in the figure 4.4. The convention of *n64s1k3* is the same convention as that explained for the Generator previously. The discriminator here has 6 convolutional layers which has shown good results in certain tasks [3]. Here the first two layers would be having 64 filters, then the next two would be having 128 filters and the last two would contain 256 filters each. Among the convolutional layers we also have added three batch normalization layers for stabilizing/optimizing the training for GAN as can be seen from the figure of the Discriminator 4.4.

Using the same methodology as the generator, for discriminator also we would have a small 3 X 3 kernel size. But here in D, we also have fully connected layers (FCL) at the end, after the last convolutional layer we would have a FCL with 1024 outputs following which we have another FCL with 512 outputs and the last FCL would have just one single output. As we are using Wasserstein GAN, following the convention from the original authors [27], we have not kept a sigmoid cross entropy layer at the end of the discriminator.

4.6 Network Training

During experiments, the model is optimized using the Adam optimizer [37]. We experimented with the model with different hyper-parameters multiple times and concluded that the values in table 4.1 for their corresponding hyper-parameters produce the best results. As we also train the slices on patches the patch size can also be considered as a hyper-parameter and in our experiments larger the size of the patch size we use larger the memory we need, thus keeping that into consideration the suggested patch size is 64 X 64.

Parameters	**Value**	**Parameters**	**Value**
Epochs	800	Perceptual loss weight	0.3
Learning rate	0.0001	Adversarial loss weight	0.5
Batch size	128	Epsilon	0.001
Discriminator's iteration	3	Adam Beta1	0.5
Gradient penalty weight	10	Adam Beta2	0.9

Table 4.1: Hyper parameters for training

Along with all this we also save checkpoints at specified interval of the model, and it also saves the final model weights for testing with different datasets. The gradient penalty weight is chosen as 10 suggested by Gulrajani et al. [29].

4.7 Training Evaluation

Once the GAN starts generating slices that are visually better than the Low-Dose CT-scans then comes the most important task of evaluating them. One of the many approaches that have also been incorporated by Yang et. el [30], is of using

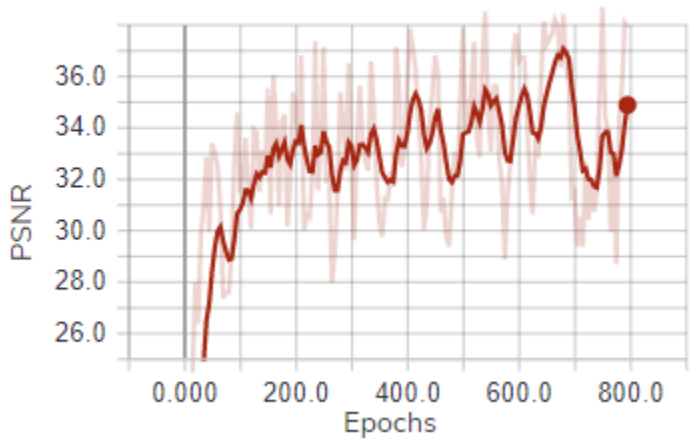


Figure 4.5: PSNR GEN vs NDCT during training

Peak Signal-to-Noise Ratio (PSNR) and Mean Squared Error Loss (MSE). Among them, PSNR is a well-known metric specifically for noise evaluations, figure 4.5 here shows the graph of PSNR while training from which we can see that the ratio is increasing with the number of epochs which is consistent with the training results obtained. This means that for each epoch the generated slices are relatively less noisy to the Low-Dose scans. It is to be noted here that the higher the value of PSNR, the lower the noise it represents. Apart from that, we also observed that the MSE values between the generated scans and the Low-Dose scans are consistent with the training results.

4.8 Handling Normalization

Initially while generating the slices from the GAN, we observed that the slices are normalized because of the normalization preprocessing performed on the input data before giving input to the model. Thus, the metrics such as MSE and PSNR were not accurate and were representing inaccurate information. So, we

subtracted the images among each other and realized that the scale for the generated image was different than that of the Low-Dose and Normal-Dose images, so to avoid this situation we performed rescaling on the generated image and added a step of post-processing to the model which will be discussed in the next section.

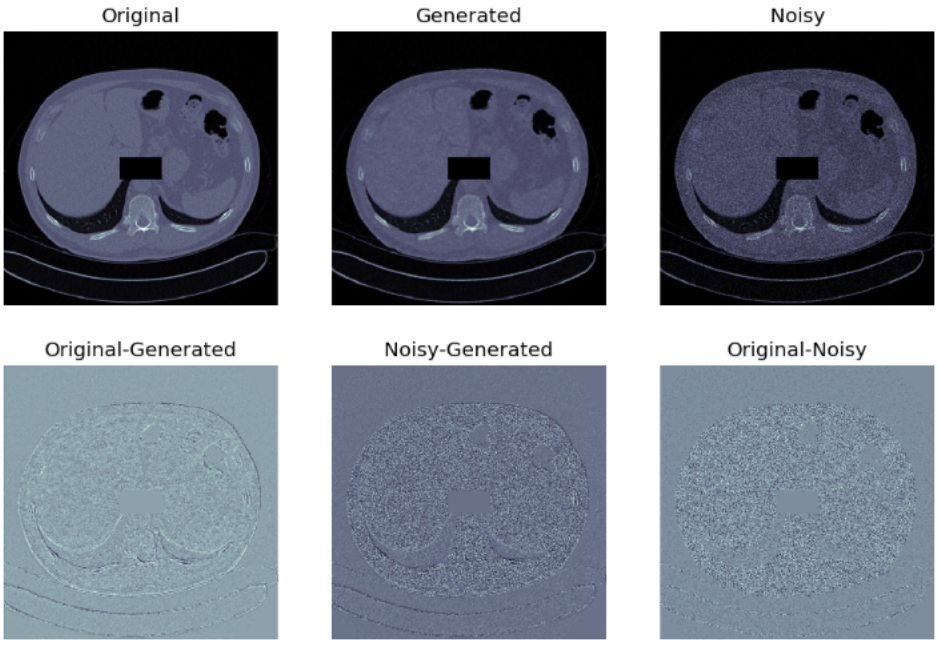


Figure 4.6: Image differences

Fig. 4.6 here shows the experiments we performed once the model was trained completely and it was producing results. We can see from the subtracted images that scale for all three of the distributions is the same which previously wasn't and thus we added the post-processing methods as mentioned before to our GAN network. We can say that all images are in the same scale because before visualising or subtracting the images we added a small block in the center of each image which we can see in the first row of the images. For the second row too the block is there but the block is visually the same in all images in the second row because subtracting zero with zero doesn't change the image pixels visually. The

only reason the second row is in gray color is because of the cmap used while actually plotting the images.

4.9 Evaluation Metric

4.9.1 PSNR and SSIM

Later in results chapter 5.2, we would be seeing the results obtained using our network and approach. As suggested by National Instruments [38], Peak signal-to-noise (PSNR) ratio can be considered as a good image quality metric. The term PSNR in itself is a ratio between the maximum power of a signal and the power of distorting noise that affects the quality [38]. Hence we use PSNR as one of the evaluation metric and for comparison as well which would be seen in later sections.

Apart from PSNR, we also use Structural Similarity Index (SSIM) as an evaluation metric. As proposed by Zhou Wang et al. [23], the SSIM metric can also be used as an image quality assessment metric. As per the equation of SSIM, we can see that it also takes the similarity of edges into account between two images. Hence it can be said that it takes the overall structure of the image also into account and not just difference between two images. SSIM is considered as a perception-based model [39] and thus is considered as an image quality assessment metric.

4.9.2 Noise Algorithm

There are many approaches to evaluate images and noise in them but since we are using CT-scans which are not just normal images we need to evaluate the scans in a manner that radiologists agree with and technicians right after the scan can understand whether the scan performed is upto the quality of that the radiologists can evaluate. Hence we use a methodology proposed by Ranjith Kamalanathan et al. [10] which focuses on fast noise and standard deviation between the regions of interests in the CT-scans. The methodology is based upon the work of Samei et al. [40] which focuses on measuring the stochastic noise.

The methodology for the evaluation first reads the CT-scan and converts the pixel values to hounsfield units so that specific regions of interest can be detected from them. The histogram with representation of the pixel values for the hounsfield units is as shown in the figure 4.7.

The histogram shown in Fig. 4.7 is generated from our original NDCT datasets for which after the preprocessing and converting the pixel values to the hounsfield units the frequency vs HU has been plotted. The table which describes the substance in the scans and corresponding expected HU is referred from [10, 40, 41]. Once the scans are loaded and necessary preprocessing is performed on the slices we perform segmentation on them and find the regions of interest for soft tissue by specifying a range of 0 to +300. The range in the table 4.2 for the soft tissue would be almost in the same range as ours but since our slices are also from the generated distribution we need to increase the range a bit. For finding the boxes

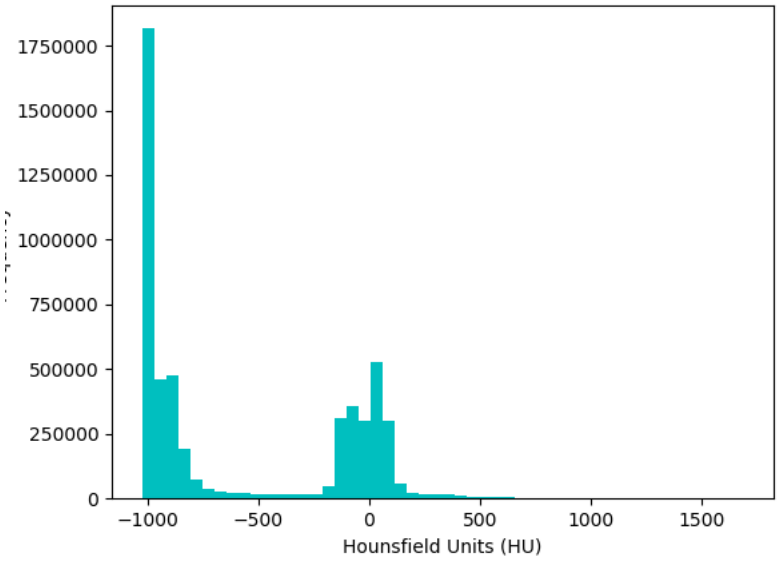


Figure 4.7: Frequency vs HU in 20 slices

that belong in the regions of interest the authors have enforced a strict bounding that is if any pixel value that does not fall in the range of 0 to +300 then we disregard that box and move on with other regions. These regions have been found using a kernel which traverses from left to right and then top to bottom. Few examples of the kernels are shown in Fig 4.8. One thing to be noted here is that the regions of interest might fall over the edges which represent the transition between

Substance	**HU**	**Substance**	**HU**
Air	-1000	Blood	+30 to +45
Lung	-500	Muscle	+10 to +40
Fat	-100 to -50	Grey Matter	+37 to +45
Water	0	White Matter	+20 to +30
CSF	15	Liver	+40 to +60
Kidney	+20 to +45	Soft Tissue	+100 to +300
Bone	+700 to +3000	Abscess/Pus	0 to +45

Table 4.2: Substances corresponding to the Hounsfield Units

two or more types of anatomical regions and thus to remove those ROIs an enhancement filter was applied to the slice image. The enhancement filter is the sobel filter [42]. Using the filter and other processing methods discussed above we were able to fetch the regions of interest from the slices.

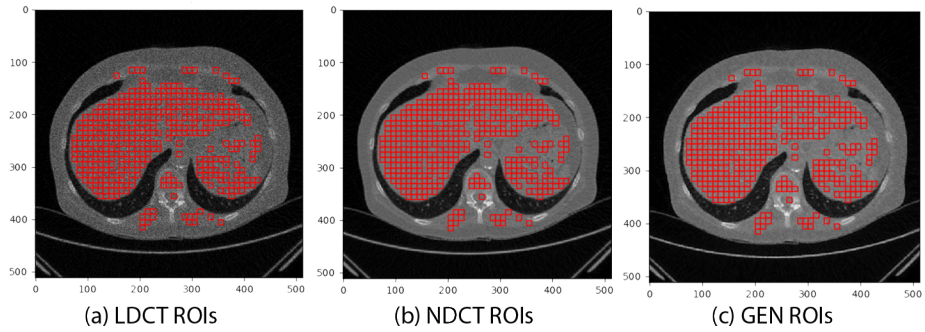


Figure 4.8: ROIs obtained from the noise algorithm

One modification that was required and performed by us to fit the algorithm for our data distribution was to store the boxes/ROIs from the NDCT slices and use the same boxes for the LDCT and generated slices so that consistent and proper noise levels are estimated for the complete data distribution. After the consolidated collection of ROIs are obtained we find standard deviation of each of the ROIs for each slice. The global noise was then calculated by the average of the Standard Deviation of all ROIs. Once we get the global noise levels, we integrate them with noise variance computed by measuring the variance across each of the pixels present in the image for the purpose of image quality analysis [10], to estimate the integration, the fast noise variance estimation method developed by John Immerkaer [43] was used.

Fig.4.8 here shows the boxes/ROIs for the slices obtained from the noise algo-

rithm. The standard deviation has been calculated over these regions and as we can see the regions are also consistent with the range specified and no region is over the other edge which would show transition between the two or more anatomical regions. We can also see the boxes obtained from the NDCT slices are also the same boxes kept on the generated and LDCT slices for better and consistent noise estimation. The results obtained from the noise algorithm is upto expectation and consistent with the experiments. More about the values and noise levels will be described in the results section.

Chapter 5: Results

5.1 Experimental Design

In this chapter we would be discussing the results obtained using the proposed CL-WGAN model. Apart from the results obtained, we also show a comparison with five other models, which also include two state-of-the-art published methods. For experimental purposes we have used the same primary dataset which includes simulated low-dose CT-scans and their corresponding normal-dose CT-scans which forms our primary dataset. The results and charts shown here are obtained from the 82 slices from 4 different patient scans, and this is the test data distribution. All models have been trained and tested on the same primary distribution, therefore the comparison is also done using the same 82 slices test data for all models. The same data distribution has also been used for obtaining the results for the Noise algorithm as shown in section 5.4.

5.2 Denoising Results

Fig.5.1 shows the final result of a test patient with specific slice. As it can be observed, the image labeled as GEN is the image generated by the generator. We

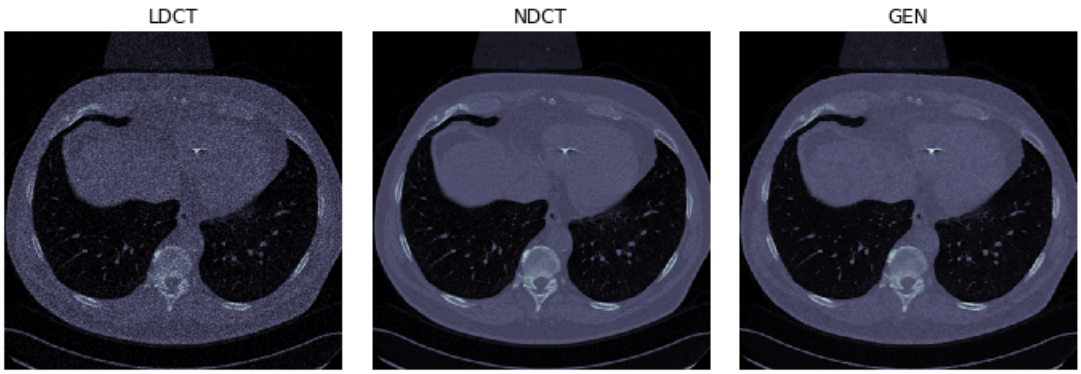
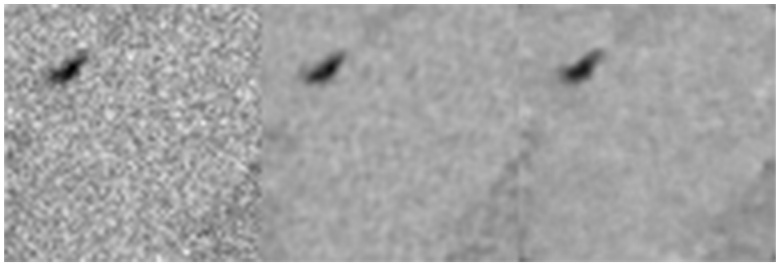


Figure 5.1: Result obtained from GAN

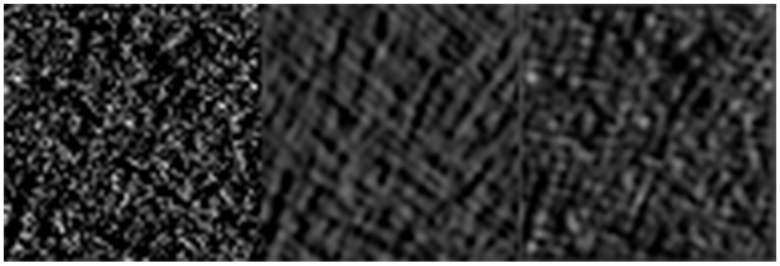
can see that compared to the simulated LDCT the generated image has significant lower amount of noise relatively. The generated image is a close representation of the NDCT ground truth image as no significant artifacts have been induced in the image and the overall structure of the image is preserved. We would be seeing the performance of the model compared with other loss functions in later sections where we would also look into the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) of the images. From the Fig.5.1 we can see that the image also preserves softer lines outside the cell structure, this is because of the use of the perceptual loss. The perceptual loss maintains the features of the ground truth image in the generator. We can see that the generated image and the ground truth NDCT image are visually similar, this is because the VGG loss/perceptual loss calculated using the VGG network is computed in a feature space that is trained previously on a very large natural image dataset [7].

It also can be seen that the amount of blur is very minimal in the generated image which means that the model actually learns to denoise the image without learning that blurring also reduces the amount of granular noise induced in the

LDCT image. The charbonnier loss [31] is very useful in improving the image quality over the number of epochs because it acts as a regularizer for our generative adversarial network.



(a) LDCT -> NDCT -> GEN



(b) LDCT -> NDCT -> GEN

Figure 5.2: Image patches created during training

Fig.4.8 can also be considered as an example of the result obtained using the generator on a test patient, the results for the noise algorithm would be discussed in the following sections. We also compare our results with certain models in the following section where PSNR, SSIM and the noise algorithm are used as the evaluation metrics.

Fig.5.2 here shows few examples of the patches that have been used for training. As mentioned previously, that we train on image patches and during training the generator generates the patches which is the right-most patch in the figure. The

left-most patch is the LDCT patch and the center patch is ground truth NDCT patch. The (b) patch in figure5.2 shows that the generator is able to reduce the granular noise from the LDCT image but the line present in the ground truth are still not obtained. Although it is not a 100 percent match we can observe that the generated patch for both (a) and (b) are visually very similar to the NDCT patch.

5.3 Quantitative Analysis and Comparison

We compare our proposed CL-WGAN model versus 5 alternative methods. Two of these methods are considered state-of-the-art CT denoising GANs from recent literature [22, 30]. We also compare against a L1 perceptual loss, the MSE perceptual loss and the adversarial loss. Finally, as a naive baseline we compare against the original noisy LDCT scan. We find that CL-WGAN achieves the highest PSNR on the test dataset of any of the methods in this comparison as seen in figure 5.3. We describe each of these methods as follows,

SSL-WGAN is the structurally senstitive loss SSIM equation as proposed by Chenyu You et al. [22]. PL-WGAN is the Wasserstein GAN architecture as proposed by [30]. WGAN is a baseline GAN using Wasserstein adversarial loss but without the use of any perceptual loss term. MSE-WGAN and L1-WGAN represents the Wasserstein GAN perceptual loss as well as the MSE perceptual loss functions respectively. LDCT represents the original test image scan with Poisson noise, and is expected to be lower than the rest of the models.

Figure 5.3 shows the PSNR ratio of the generated images for 82 test slices for

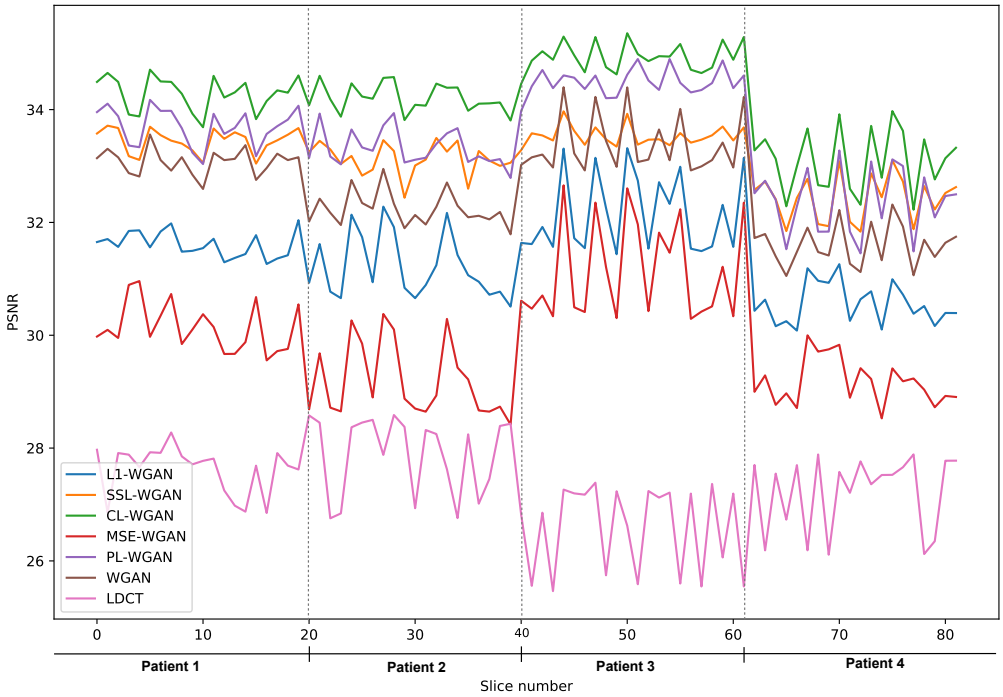


Figure 5.3: PSNR obtained during testing

each model in the comparison. We can see that CL-WGAN was able to produce the highest PSNR when compared to other methods, followed by PL-WGAN [30], SSL-WGAN [22], and WGAN. The MSE-WGAN and L1-WGAN achieved a lower PNSR.

Figure 5.4 shows the comparison of the Structural Similarity Index (SSIM) which is a perceptual metric for image quality. We see that the proposed CL-WGAN outperforms the other methods in accordance with this metric as well. Overall, the ordering of the results with SSIM are largely similar to the PSNR results, although we see that WGAN performs more competitively in comparison to the PL-WGAN [30] and SL-WGAN [22] models for this evaluation metric.

It can be seen that although the MSE-WGAN achieves relatively higher PSNR ratio the SSIM value for the same is very low compared to others and in some cases

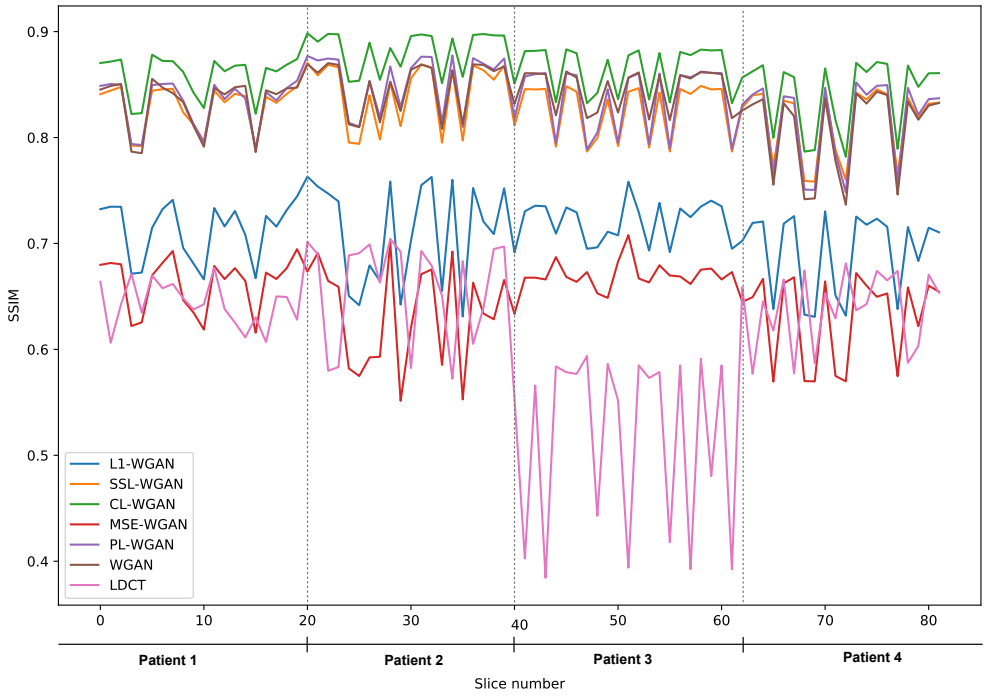


Figure 5.4: SSIM obtained during testing

very close to the LDCT curve as well, this can be considered as a disadvantage of using MSE Loss with the GAN because as mentioned previously and shown by many researchers [9, 30] it induces blur in the generated images which in turn can add unnecessary artifacts and reduce the overall structural composure of the image.

5.4 Results obtained using the Noise Algorithm

The results obtained and calculated using the Noise Algorithm proposed by Ranjith Kamalanathan et al. [10] are consistent with the quantitative results obtained using PSNR and SSIM as shown in the previous sections.

In table 5.1 we see the results obtained by passing figure 5.1 to the noise algorithm proposed by Ranjith Kamalanathan et al. [10] which is largely similar to

	SD Noise
LDCT	169.41
NDCT	42.93
CL-WGAN	46.13
SSL-WGAN	47.01
PL-WGAN	52.42
MSE-WGAN	51.90
L1-WGAN	51.43
WGAN	41.96

Table 5.1: Noise Algorithm results on figure 5.1

the method developed by Ehsan Samei [40]. The noise algorithm is discussed in more detail in section 4.9 on page 38. Essentially here SD Noise represents the value of standard deviation in soft tissue and other smooth regions within the scan.

In an ideal scenario, a denoising algorithm should achieve the noise estimation using this approach which is as close as possible to the NDCT ground truth slices. As it can be seen in table 5.1, the NDCT can be considered as the value that we are trying to achieve and compared to other algorithms CL-WGAN provides a value that is closest to the NDCT image. The LDCT slice does contain simulated noise levels and thus the value provided by the algorithm is extremely high which is as expected. We can see that although SSL-WGAN does not provide close results to us relatively for PSNR and SSIM it does maintain structural integrity and reduces noise better than other methods as the value is relatively low with respect to other comparisons. Although the difference between PL-WGAN and MSE-WGAN is not much it is safe to say that PL-WGAN provides visually better quality results than MSE-WGAN as per our observation on test slices and also

the noise algorithm did gave lower values than MSE-WGAN for most of the slices. The PSNR and SSIM obtained for PL-WGAN is also higher than that of MSE-WGAN which provides validity to the generated slices.

Apart from that we also show the results obtained for the L1-WGAN and WGAN here, which are the networks as discussed in the previous section. The L1-WGAN here produces very close results to MSE-WGAN and understanding the fact that L1 essentially calculates a difference between two images the relative closeness with MSE-WGAN is to be expected. Also, WGAN here is seen to be producing values lower than that of the ground truth image which means that the generated images from only using the adversarial loss are over-smoothed which also supports our visual observation. And although for some slices as per our observation WGAN estimates the closest to NDCT, the overall mean values are smaller relatively to the other models which suggests that it compromises the information and the structure of the slices [30].

Following the discussion, we would now show more results of the test data slices obtained from the generator and evaluated on the noise algorithm. It is also to be noted that all the images shown in the above sections and the following images contribute to the PSNR 5.3 and SSIM 5.4 plots as well.

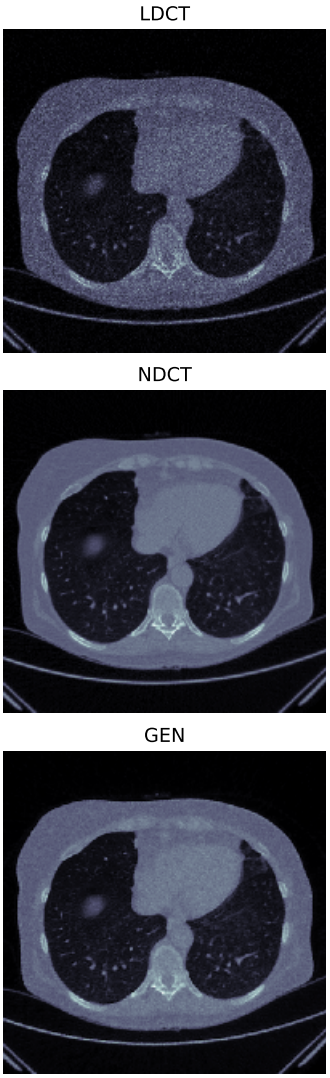


Figure 5.5: Result 2

	SD Noise
LDCT	168.30
NDCT	38.59
CL-WGAN	46.52
SSL-WGAN	47.10
PL-WGAN	52.95
MSE-WGAN	52.51
L1-WGAN	52.61
WGAN	42.51

Table 5.2: Noise values for 5.5

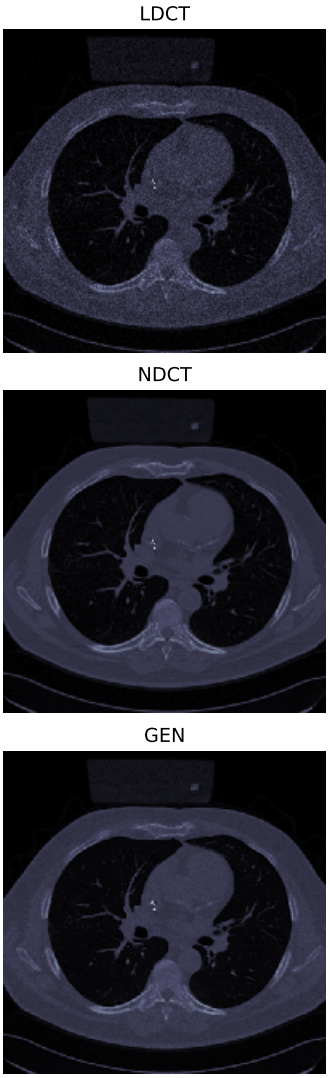


Figure 5.6: Result 3

	SD Noise
LDCT	167.74
NDCT	38.29
CL-WGAN	45.54
SSL-WGAN	46.86
PL-WGAN	52.36
MSE-WGAN	51.52
L1-WGAN	50.95
WGAN	41.45

Table 5.3: Noise values for 5.6

Chapter 6: Conclusion

The main goal and motivation for this paper is to achieve the ground truth NDCT image given a Low-Dose CT-scan. There are many ways to build up the mapping from LDCT to NDCT and capture the noisy features and denoise them to generate a close representation of the NDCT scans. This research hence provides a thorough evaluation of a network model which is more dedicated towards combining synergistic loss functions such as the perceptual loss and the charbonnier loss to guide the denoising process, so that the resultant denoised slices are as close as to their corresponding NDCT slices.

The perceptual loss in the model provides us with a closer understanding of human perception which is embedded in the VGG network as it is previously trained on a large natural image dataset. This human perception proves to be important when compared the results along with the MSE loss metric, we can see that the results obtained using the perceptual loss are better than that obtained from MSE loss visually as well. We can also agree that using just Wasserstein GAN or a GAN alone would not be able to provide the results currently obtained as a GAN only provides the map of the data distribution between LDCT and NDCT and not the image content correspondence which is of high important for CT-scan

evaluation.

The charbonnier loss along with network optimization for faster training proves to be a good regularization function for the Wasserstein GAN during training. As it also is a variant of Huber loss and known as pseudo-Huber loss, it acts as a regularizer to train the GAN and produce images with an approximate quality as the NDCT (ground truth) images.

Hence, we conclude that using a combination of loss functions that are targeted towards the main goal of achieving denoised CT-scans produces better results quantitatively which are consistent with the evaluation on a published noise evaluation metric. We also show that the denoising of CT-scans can be achieved by using the appropriate loss functions and does not always need the network architecture to be complex. Therefore, we can say that our network architecture along with the integration of the Perceptual loss and Charbonnier Loss produces images with reduced level of noise and can be said as a close representation of the Normal-Dose CT-scan.

Chapter 7: Future Work

As we discussed in chapter 2, there are several ways to utilize the reconstruction methods which focus more on the statistical side of image production and provides denoised images. One thing to extend our current model and provide better evaluation is to compare the results with some of the reconstruction networks and algorithms along with some new metrics such as quality of the image or the time invested in generating a image given a low-dose CT-scan. One of the other extension to current model can be of building a complex network architecture such as using residual networks/skip connection layers in the GAN or use cyclic loss function based on the residual layers and integrate the same with the Perceptual loss and Charbonnier loss functions used currently. Few complex networks as such as discussed in the related work section as we have seen previously. More complicated generators may can improve the results or degarde it but it for sure can be considered as a future experimentation in addition to the current model.
In addition to making the model a bit complex, we can also train the model or even test the robustness of the model by using multiple datasets. Having a large amount of data coming from different places but with same distribution can help the model be robust because different datasets can have different amount of noise

in them and thus the model gets a chance to learn denoising and noisy features of varying amounts. This can be helpful in making a complete robust denoising model which can generate a denoised image which is a close approximate of the NDCT image from a noisy LDCT image containing noise of different levels.

As we saw from the results described in 5.3, 5.4 and 5.1, changing the loss functions and network architecture can produce different results for denoising networks and thus using an appropriate additive loss function can be a crucial measure of improving the model. More experimentation can be performed on extending the equation 4.1 by adding/removing different loss functions that can generate denoised slices. Novel loss function which provides a reasonable insight in denoising images can also be integrated with this model and can be made more robust to noise in not only CT-scans but natural images as well. Since denoising in CT-scans includes maintaining the image's overall structure and the content the same can also be probably used for denoising natural images which contain a similar noise levels as the Low-dose CT-scans.

Bibliography

[1] Lee W Goldman. Principles of ct: radiation dose and image quality. *Journal of nuclear medicine technology*, 35(4):213–225, 2007.

[2] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Neural photo editing with introspective adversarial networks. *arXiv preprint arXiv:1609.07093*, 2016.

[3] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[4] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[5] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[8] Marcel Beister, Daniel Kolditz, and Willi A Kalender. Iterative reconstruction methods in x-ray ct. *Physica medica*, 28(2):94–108, 2012.

[9] Jelmer M Wolterink, Tim Leiner, Max A Viergever, and Ivana Išgum. Generative adversarial networks for noise reduction in low-dose ct. *IEEE transactions on medical imaging*, 36(12):2536–2545, 2017.

[10] Ranjith Kannan Kamalanathan. Automated techniques for measuring and predicting clinical computed tomographic images. 2019.

[11] Davood Karimi, Pierre Deman, Rabab Ward, and Nancy Ford. A sinogram denoising algorithm for low-dose computed tomography. *BMC medical imaging*, 16(1):11, 2016.

[12] Armando Manduca, Lifeng Yu, Joshua D Trzasko, Natalia Khaylova, James M Kofler, Cynthia M McCollough, and Joel G Fletcher. Projection space denoising with bilateral filtering and ct noise modeling for dose reduction in ct. *Medical physics*, 36(11):4911–4919, 2009.

[13] Bruce R Whiting, Parinaz Massoumzadeh, Orville A Earl, Joseph A O'Sullivan, Donald L Snyder, and Jeffrey F Williamson. Properties of preprocessed sinogram data in x-ray computed tomography. *Medical physics*, 33(9):3290–3303, 2006.

[14] Idris A Elbakri and Jeffrey A Fessler. Statistical image reconstruction for polyenergetic x-ray computed tomography. *IEEE transactions on medical imaging*, 21(2):89–99, 2002.

[15] Sathish Ramani and Jeffrey A Fessler. A splitting-based iterative algorithm for accelerated statistical x-ray ct reconstruction. *IEEE transactions on medical imaging*, 31(3):677–688, 2011.

[16] Emil Y Sidky and Xiaochuan Pan. Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization. *Physics in Medicine & Biology*, 53(17):4777, 2008.

[17] Qiong Xu, Hengyong Yu, Xuanqin Mou, Lei Zhang, Jiang Hsieh, and Ge Wang. Low-dose x-ray ct reconstruction via dictionary learning. *IEEE transactions on medical imaging*, 31(9):1682–1697, 2012.

[18] Wei Hua and Youshen Xia. Low-light image enhancement based on joint generative adversarial network and image quality assessment. In *2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 1–6. IEEE, 2018.

[19] Xin Yi and Paul Babyn. Sharpness-aware low-dose ct denoising using conditional generative adversarial network. *Journal of digital imaging*, 31(5):655–669, 2018.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. corr abs/1512.03385 (2015), 2015.

[21] Chenyu You, Linfeng Yang, Yi Zhang, and Ge Wang. Low-dose ct via deep cnn with skip connection and network in network. *arXiv preprint arXiv:1811.10564*, 2018.

[22] Chenyu You, Qingsong Yang, Lars Gjesteby, Guang Li, Shenghong Ju, Zhuiyang Zhang, Zhen Zhao, Yi Zhang, Wenxiang Cong, Ge Wang, et al. Structurally-sensitive multi-scale deep neural network for low-dose ct denoising. *IEEE Access*, 6:41839–41855, 2018.

[23] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[24] Eunhee Kang, Hyun Jung Koo, Dong Hyun Yang, Joon Bum Seo, and Jong Chul Ye. Cycle-consistent adversarial denoising network for multiphase coronary ct angiography. *Medical physics*, 46(2):550–562, 2019.

[25] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[26] Lilian Weng. From gan to wgan. *arXiv preprint arXiv:1904.08994*, 2019.

[27] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[28] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[29] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[30] Qingsong Yang, Pingkun Yan, Yanbo Zhang, Hengyong Yu, Yongyi Shi, Xuanqin Mou, Mannudeep K Kalra, Yi Zhang, Ling Sun, and Ge Wang. Low-dose ct image denoising using a generative adversarial network with wasserstein distance and perceptual loss. *IEEE transactions on medical imaging*, 37(6):1348–1357, 2018.

[31] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceedings of 1st International Conference on Image Processing*, volume 2, pages 168–172. IEEE, 1994.

[32] Alice Lucas, Santiago Lopez-Tapia, Rafael Molina, and Aggelos K Katsaggelos. Generative adversarial networks and perceptual losses for video super-resolution. *IEEE Transactions on Image Processing*, 28(7):3312–3327, 2019.

[33] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992.

[34] Jonathan T Barron. A general and adaptive robust loss function. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4331–4339, 2019.

[35] Data science bowl 2017. https://www.kaggle.com/c/data-science-bowl-2017/data.

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

[37] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[38] Peak signal-to-noise ratio as an image quality metric. https://www.ni.com/en-us/innovations/white-papers/11/peak-signal-to-noise-ratio-as-an-image-quality-metric.html.

[39] Structural similarity. https://en.wikipedia.org/wiki/Structural_similarity.

[40] Olav Christianson, James Winslow, Donald P Frush, and Ehsan Samei. Automated technique to measure noise in clinical ct examinations. *American Journal of Roentgenology*, 205(1):W93–W99, 2015.

[41] Hounsfiled scale. https://en.wikipedia.org/wiki/Hounsfield_scale.

[42] Nick Kanopoulos, Nagesh Vasanthavada, and Robert L Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of solid-state circuits*, 23(2):358–367, 1988.

[43] John Immerkaer. Fast noise variance estimation. *Computer vision and image understanding*, 64(2):300–302, 1996.