PES University

(Established under Karnataka Act No. 16 of 2013)
100 Feet Ring Road, Bengaluru, Karnataka, India – 560 085

Project Report

Jan - May 2020

Denoising of Low Dose CT Images Using Generative Models

Lalithnarayan C - 01FB16EEC139

Manik B Somayaji - 01FB16EEC149

Azhar Shaikh - 01FB16EEC370

Under the guidance of

Dr. Chethan KS
Assistant Professor
Department of Electronics and Communication
Engineering
PES University
Bengaluru - 560085

FACULTY OF ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

PROGRAM B. TECH.

CERTIFICATE

This is to certify that the Report entitled

Denoising of Low Dose CT Images Using Generative Models

is a bonafide work carried out by

Lalithnarayan C (01FB16EEC139)

Manik B Somayaji (01FB16EEC149)

Azhar Shaikh (01FB16EEC370)

In partial fulfillment for the completion of 8th semester course work in the Program of Study B. Tech. in Electronics and Communication Engineering, under rules and regulations of PES University, Bengaluru during the period Jan – Apr 2020. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 8th semester academic requirements in respect of project work.

Signature with date & Seal	*Signature with date & Seal*	*Signature with date & Seal*
Dr.Chethan K S	*Dr. Anuradha M*	*Dr. Keshavan BK*
Internal Guide	*Chairperson*	*Dean of Faculty of Engineering*

Name and signature of the examiners:

1.

2.

DECLARATION

We, **Lalithnarayan C**, **Manik B Somayaji** and **Azhar Shaikh**, hereby declare that the project report entitled, ***Denoising Of Low Dose CT Images Using Generative Models,*** is an original work done by us under the guidance of **Dr.Chethan KS**, Designation, Dept. of ECE and is being submitted in partial fulfillment of the requirements for completion of 8th Semester course work in the Program of Study B. Tech. in Electronics and Communication Engineering.

PLACE:
DATE:

NAME AND SIGNATURES OF THE CANDIDATES

1. Lalithnarayan C
2. Manik B Somayaji
3. Azhar Shaikh

Abstract

Medical image denosing is a critical step in the preprocessing of data acquired through medical imaging methods like Computational Tomography(CT) and Magnetic Resonance Imaging(MRI). The quality of the medical image generated depends on the dosage levels of the radiation used. The higher the radiation the better the penetration through the tissues, and thus better images are obtained. The disadvantage with this approach is exposing the patient to various medical hazards due to high radiations. Therefore, we approach the problem of denosing of low-dose CT scan images using various generative models such as the Auto-encoder based RED-CNN, and the GAN based Wasserstein-GAN and Pix2Pix GAN. The project involves understanding the generative models in detail and comparing the three models on the problem of denoising of images. The results obtained are tested using both quantitative measures as well as qualitative measures.

Acknowledgement

The project has been possible through the guidance and timely help of many people. In this section, we would like to acknowledge their help and sincerely thank them for their support. We would first like to thank Dr. Chethan KS for guiding us through the entire process. We thank Dr. Chandar TS and Ms. Sarita Uniyal for their important inputs during the monthly review meetings.

We would like to thank the doctors who agreed to support us by agreeing to review the project. We would like to thank Dr. Sudheendra,Chief Consultant Radiologist and Dr. Shailendra, Consultant Radiologist for their reviews, suggestions and guidance. The guidance has concretely motivated us to improve the current state of the work into a commercially viable product.

We would like to thank Mr. Akash Narayana, Senior Data Architect, RX Digi Health Platform Pvt Ltd, for getting us in touch with Dr.Shailendra. We would like to thank Mr. M N Chandrashekar and Mr. S.M. Hukkeri for helping us contact Dr. Sudheendra. The help offered smoothened the process.

Last but not least, we would like to thank our parents for their constant and encouraging support. We would also thank the PES University administration for the smooth conduct of the entire project phase during the lockdown.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Medical image denoising is a critical step in the processing of medical image data that is acquired through methods such as Computational Tomography(CT) and Magnetic Resonance Imaging(MRI). The noise present in the image hinders the ability of a radiologist to efficiently and quickly analyse the medical data and provide critical insights to doctors who diagnose patients. The clarity of image produced depends on the dose of the radiation used by a medical imaging machine, for example, SIEMENS SOMATOM Series. Higher the radiation dosage setting of the machine, clearer and crisp is the image recorded. However, the higher dose of radiation also puts the health of the already sick patient at risk. To mitigate the harmful effect of radiation, medical image data is collected with low dose radiation setting with the now added drawback of having to work with noisy and unclear medical image data. This work primarily deals with denoising of CT scan data but the methods explored can be extended to other forms of medical image data. From here on we will refer to Low dose CT data as LDCT and normal dose CT data as NDCT.

1.2 Problem Statement

Finding a generative functional transformation from a LDCT image domain to a NDCT image domain that preserves information about the underlying medical structures in the LDCT domain while producing an image in the NDCT domain devoid of any noise and artifacts. Deep generative models like Auto-encoders and Generative Adversarial Neural Networks (GANs) are viable methods given their success on Denoising and Super Resolution tasks on natural images. In this project we answer the viabilty of the generative models on the task of denoising of LDCT images.

1.3 Motivation

The field of generative models is advancing at a very rapid speed, thanks to the huge research efforts put in by the research community. Therefore, we recognize this as an opportunity to apply these models to solve the problem of denoising of medical images and understand the applicability of generative models for medical data in depth. We intend to solve many problems through the project. They are

- Reduce health risks for the patient.
- Reduce the carbon footprint involved with the generation of CT Scans.
- Reduce the cost associated with the power usage by CT scan machines.
- Understand generative models and hopefully implement more of these in the future.

The added advantage of generative models is that they learn the joint probability function. This enables them to scale versus discriminatory models which model the conditional probability. Therefore, these models are only as good as the data they have been trained on.

1.4 Outline

In the following report, we have various chapters focusing on major key areas of our project. In Chapter 2 we consider the various approaches taken so far. In Chapter 3, we discus the basic concepts required to implement the models mentioned. These basics are the fundamentals of deep learning, and a brief overview of the same would help the reader. The chapters 4 and 5 discuss the details about the dataset used, dataset acquisition procedure and the dataset preprocessing required. Chapters 6, 7 and 8 discuss the three generative models we have implemented in the project, in detail. The results of the various models have been discussed in the 9^{th} chapter. Finally we present the doctor review and our take on the results.

Chapter 2

Literature Survey

Extensive efforts are being made in the field of medical imaging especially in the research area of finding better methods for the reconstruction of X-ray computed tomography for low dose radiation exposure. The methods are broadly classified in three groups:

1. Sinogram filtration before reconstruction
2. Iterative reconstruction
3. Image post processing after reconstruction

We further classify these methods into two approaches

2.1 Traditional Approach

2.1.1 Sinogram filtration before reconstruction

A sinogram is a special x-ray procedure that is done to visualize any abnormality in the body, following the injection of contrast media (x-ray dye). This method and the post-processing method which we are going to discuss later are computational more efficient than the iterative reconstruction. The noise characteristics is well modeled in the sinogram-domain filtration. The sinogram data of commercial scanners are not readily available to users or researchers and these methods may suffer from resolution loss and edge blurring. So Sinogram data need to be carefully processed, otherwise artifacts may be induced in the reconstructed images.

2.1.2 Iterative reconstruction

Researchers have dedicated lot of time developing methods falling in this group. These algorithms generally try to optimise the objective function which incorporate system model and statistical noise model. Example dictionary learning.

These algorithms are successful in enhancing the medical image quality but still

lacks the complexity to retain the structural details which is very important for the diagnosing purpose. Also these algorithms have a high computing cost which is a bottle neck in practical applications

2.2 Deep Learning Approach

2.2.1 Image post processing after reconstruction

This method is completely different than the other two methods because unlike the sonogram pre-filtering and iterative reconstruction this method acts directly on the the generated image. Many algorithms were used in image domain to reduce the Low-Dose CT noise and suppress artifacts. For example:

1. The non-local means (NLM) method was adapted for CT image de-noising. This method unlike the local mean filters which take average of the fixed number of pixels around it, takes mean of all pixels in the image, weighted by how similar these pixels are to the target pixel. This results in greater clarity and less loss of structural details than the local mean approach.

2. With the growth of compressed sensing methods, an adapted K-SVD method was proposed to reduce artifacts in CT images.

3. The block-matching 3D (BM3D) algorithm was used for image restoration in several CT imaging tasks consists of grouping similar image fragments using clustering methods like k-means followed by performing collaborative filtering on the groups, the only caveat is that the grouped fragments are not necessarily disjoint.

With such image post-processing,image quality improvement was clear but over-smoothing and/or residual errors were often observed in the processed images. These issues are difficult to address, given the non-uniform distribution of CT image noise.

Chapter 3

Deep Learning Fundamentals

3.1 Introduction

Deep learning is a sub-field of machine learning inspired by the way learning in the human brain works, The function of the brain is dictated by the interconnection of thousands of neurons. These neurons are modelled as an artificial neural network that is capable of learning a specific task given its training examples. Due to the availability of large data resources and compute power, deep learning has emerged to be the state of the art for solving various challenging tasks like object detection, machine translation, speech generation etc.

3.2 Deep Neural Networks

The goal of a deep neural network[1] is to approximate some function f^*. For Example a regression model $y = f^*(x)$ maps input $x \epsilon R^d$ to a value $y \epsilon R^1$. A deep neural network defines a mapping

$$y = f^*(x, \theta)$$

and learns the parameters θ that produces the best approximation.

The building block of a neural network is a neuron, each neuron is connected to all the inputs x_i through synaptic connections called network weights W_i and bias b. The output of the neuron is the linear combination of the inputs and the weights plus the bias b. This output is then passed through an activation function like a sigmoid or tanh to introduce non-linearity in the network.

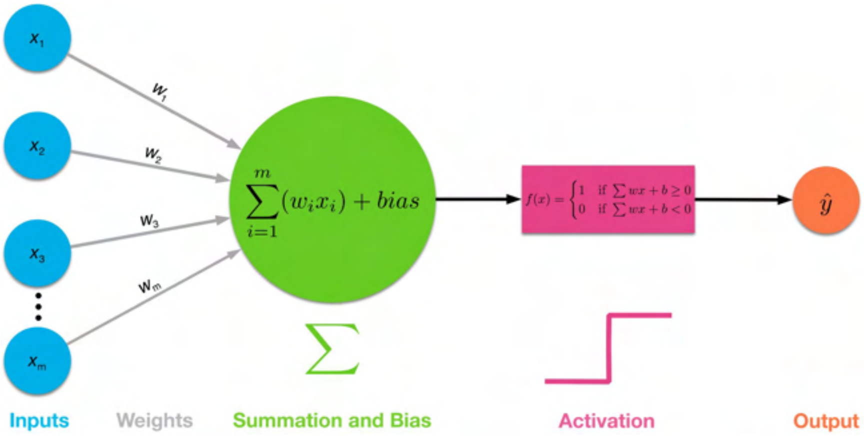

x_1
x_2
x_3
x_m
w_1
w_2
w_3
w_m
$\sum_{i=1}^{m}(w_i x_i) + bias$
$f(x) = \begin{cases} 1 & \text{if } \sum wx + b \geq 0 \\ 0 & \text{if } \sum wx + b < 0 \end{cases}$
$\hat{y}$
$\sum$
Inputs
Weights
Summation and Bias
Activation
Output

Figure 3.1: Mathematical Model of a Neuron

A Neural Network consists of an input layer, output layer and hidden layer. The hidden layer is built by stacking the neurons also called hidden units, The neurons of each hidden layer are fully connected to neurons of the previous layers i.e the activations of the previous hidden layer serves as input to the current hidden layer. For example we might have a network with three hidden layers $f^{(1)}$,$f^{(2)}$,$f^{(3)}$ and input x then the output at the output layer is

$$y = f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x)))$$

The depth of the network is equal to the number of hidden layers present in the network. Deep neural networks usually contain more than 100 hidden layers hence the name deep learning. It is this structural property of neural networks that helps them learn feature hierarchies with features from higher levels of the hierarchy formed by the composition of lower level features. Feature hierarchies help with good representation of the input data to learn complex functional mappings from the input to the output. Unlike traditional machine learning that require hand-crafted features, deep neural networks automatically learns the best features to solve a given task.

3.3 Cost Function and Optimisation

The cost function J(θ,x) or objective is a differentiable function of the parameters of the network and the input data x and its value gives an indication of whether the model parameterised by is learning or not. The cost function is minimised with respect to the parameters in most cases using different iterative optimisation methods. The selection of a loss function depends on the problem at hand. For example the loss function used for binary classification is the binary cross-entropy loss given by

$$J(\theta, x) = -(ylog(p_\theta(x)) + (1-y)log(1-p_\theta(x)))$$

and the loss for a regression model is the L2 or mean squared error loss given by

$$J(\theta, x) = \frac{1}{2}\sum_{n=1}^{N}(y(x_n;\theta) - t_n)^2$$

An analytical solution to find the parameters of a model is not tractable. Therefore unconstrained iterative first order or second order optimisation methods are used. The most popular of these optimisation methods is Stochastic Gradient Descent(SGD), SGD updates the parameters of the network in the opposite direction of the gradient of the cost function iteratively in small steps of α to reach a minima in the parameter space. α is called the learning rate and its value decides the speed of learning of the network, too small a value of α longer it takes to converge to a local minima and too large a value of α leads to unstable learning with oscillations in the value of the cost function. α is a hyper parameter and its value is selected through trial and error. The parameter update rule is given by the equation

$$\theta_i = \theta_{i-1} + \alpha\nabla J(x, \theta_{i-1})$$

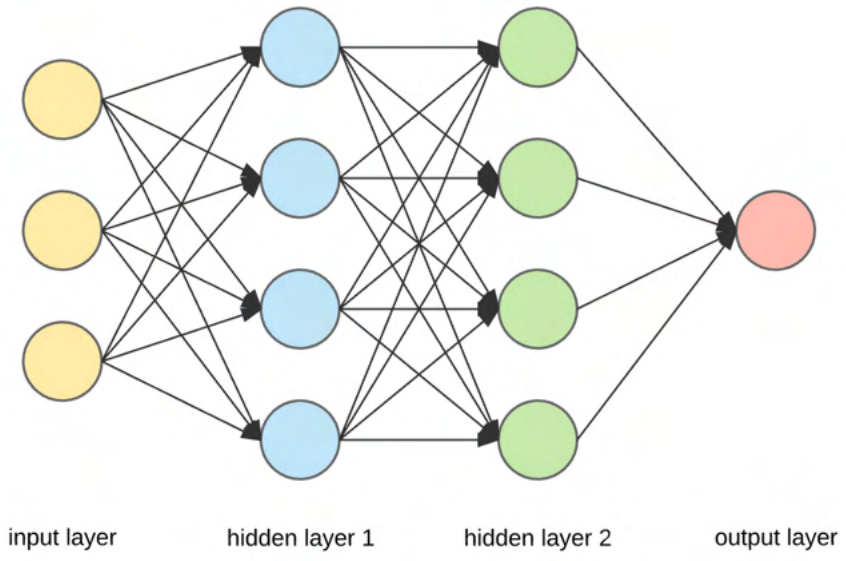


Figure 3.2: A Simple Neural Network

3.4 Convolutional-Neural Networks

Convolution Neural Network(CNN) is a special kinds of neural network architecture that makes the assumption that the input to the network is an image thus enabling us to encode certain information into the network that makes visual tasks easy to solve. In normal neural networks each neuron is fully connected to the neurons of the previous layers, this property of neural networks doesn't let them scale well to high dimensional input like high resolution images. CNNs work very well for image data and are the current state of the art models for visual tasks. They treat the input as a 3D volume and therefore have neurons arranged in 3 dimensions: width, height, depth. You can think of a CNN as a function that transforms a 3D input volume to a 3D output volume. Each neuron in the output layer is connected to a small portion of the input volume. The CNN is made up of a special set of layers like the convolutional layer, ReLU activations, pooling layers and fully-connected layers. The convolutional layer's parameters consist of a set of learnable filters. Every filter is small spatially but extends through the full depth of the input volume. During the forward pass, we convolve each filter across the width and height of the input volume and compute dot products between the entries of the filter and the input at any position. As we slide the filter over the width and height of the input volume we will produce a 2-dimensional activation map that gives the responses of that filter at every spatial position. Each filter will produce a separate 2-dimensional activation map. We stack these activation maps along the depth dimension and produce the output volume. ReLU is an ativation function that is commonly used in CNNs, The max pooling layer is used to decrease the spatial size of the input volume.

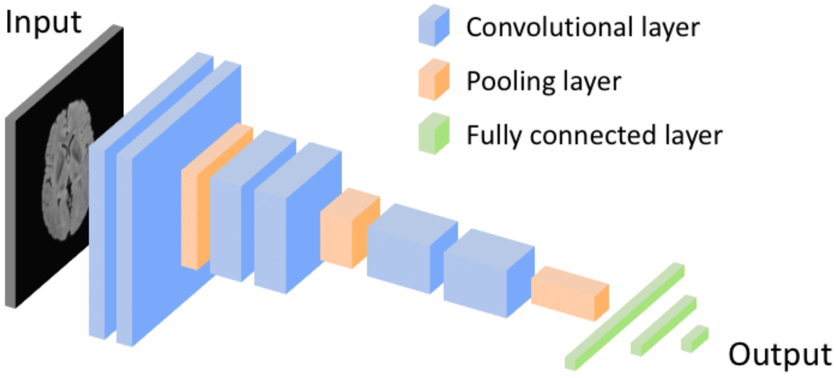


Figure 3.3: A Convolutional Neural Network

Chapter 4

Dataset

4.1 About the Dataset

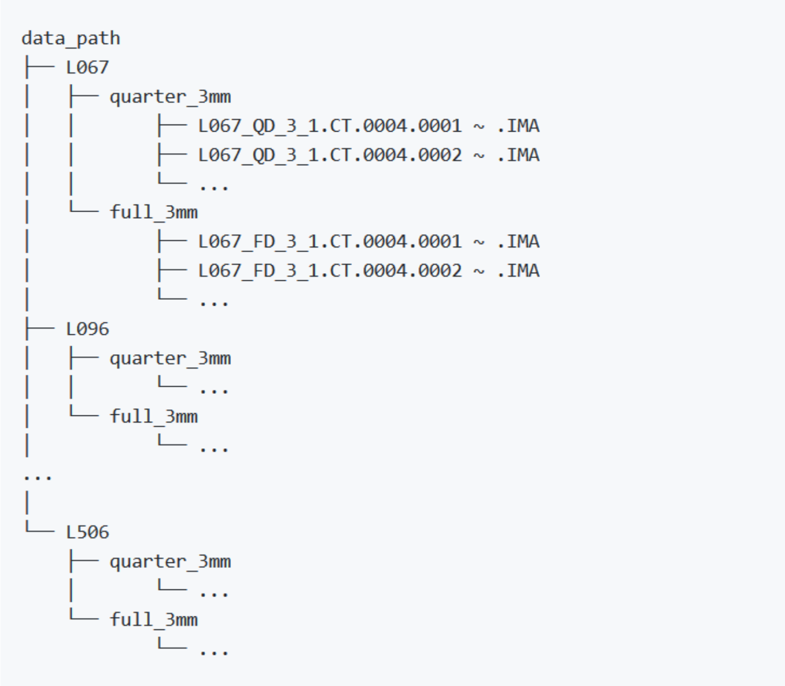
The aim of our project is to post process the low dose CT scan image such that it meets the quality of a Normal dose CT scan image. So for this project we have used the real clinical dataset which was released for “the 2016 NIH-AAPM-Mayo Clinic Low Dose CT Grand Challenge” by Mayo Clinic[2]. The dataset contains low dose CT scan images and corresponding Normal dose CT scan images of 10 anonymous patients. The low dose CT scan image is a simulated quater dose CT images by introducing noise to NDCT images. The noise is modeled to be a combination of Gaussian electronic noise and Poisson quantum noise.

4.1.1 Information Regarding Reconstruction Kernels

Each patient has his/her own zip folder. Within each zip folder there are projection data, 3 mm images and 1 mm images for full dose and quarter dose (6 data sets total). In our project we use 3mm images and convert it to numpy files. We extract random 80 ×80 patches from each numpy file and if it does not contain much information then it will be rejected. We are using 80 ×80 patches instead of full images to train because of the following reasons:

1. The generative network which we are using are data hungry and require large datasets. Especially medical dataset is hard to procure and limited in avialability.
2. Another justification for patch wise training is the model will be trained for any CT scan image (i.e any body part CT scan). The model will be prepared for more a general dataset.

The folder structure of the Dataset is shown below:



```
data_path
├── L067
│    ├── quarter_3mm
│    │       ├── L067_QD_3_1.CT.0004.0001 ~ .IMA
│    │       ├── L067_QD_3_1.CT.0004.0002 ~ .IMA
│    │       └── ...
│    └── full_3mm
│            ├── L067_FD_3_1.CT.0004.0001 ~ .IMA
│            ├── L067_FD_3_1.CT.0004.0002 ~ .IMA
│            └── ...
├── L096
│    ├── quarter_3mm
│    │       └── ...
│    └── full_3mm
│            └── ...
...
│
└── L506
     ├── quarter_3mm
     │       └── ...
     └── full_3mm
             └── ...
```

Figure 4.1: Folder structure of Dataset

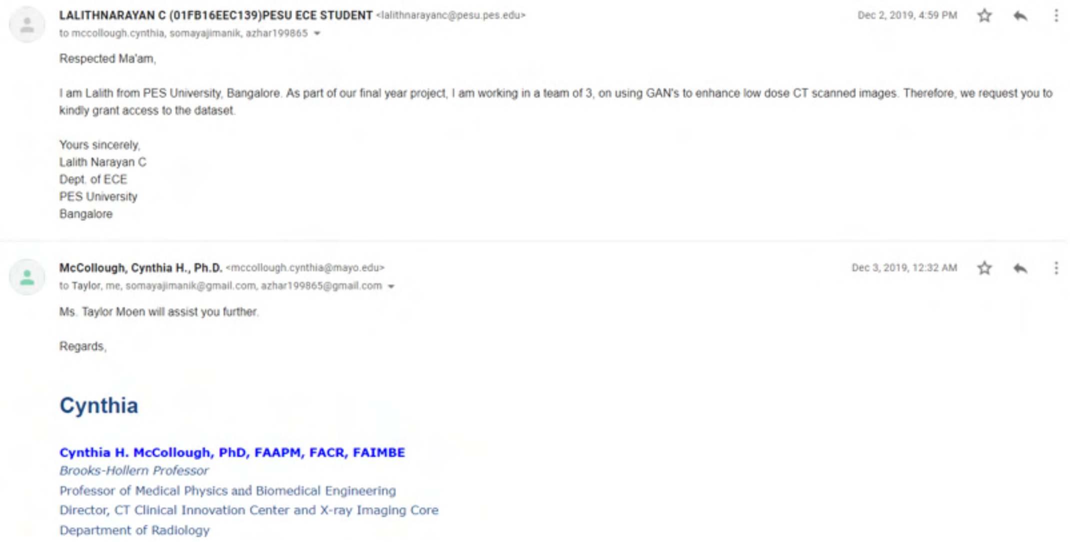
The images in each case's folder were reconstructed with a medium smooth kernel. There is another folder named "sharps" that contains additional 3 and 1 mm images reconstructed with a medium sharp kernel.

Kernel Information

B30 – this is the typical kernel used for interpreting liver CT cases in our practice. It has a mild overshoot in MTF in the lower frequencies. D45 – this is a sharp quantitative kernel, with no overshoot in the MTF, which is used for coronary stent and lung imaging.
The MTF characteristics of these kernels are provided below:

(values given in 1/cm)	**MTF value at 50%**	**MTF value at 10%**	**MTF value at 2%**
B30 kernel	3.6	5.9	7.3
D45 kernel	5.6	item 9.4	11.4

The image noise (standard deviation of CT numbers in a region of interest) is increased dramatically (by approximately 128%) when using D45 vs. B30.



LALITHNARAYAN C (01FB16EEC139)PESU ECE STUDENT <lalithnarayanc@pesu.pes.edu>
to mccollough.cynthia, somayajimanik, azhar199865

Dec 2, 2019, 4:59 PM

Respected Ma'am,

I am Lalith from PES University, Bangalore. As part of our final year project, I am working in a team of 3, on using GAN's to enhance low dose CT scanned images. Therefore, we request you to kindly grant access to the dataset.

Yours sincerely,
Lalith Narayan C
Dept. of ECE
PES University
Bangalore

McCollough, Cynthia H., Ph.D. <mccollough.cynthia@mayo.edu>
to Taylor, me, somayajimanik@gmail.com, azhar199865@gmail.com

Dec 3, 2019, 12:32 AM

Ms. Taylor Moen will assist you further.

Regards,

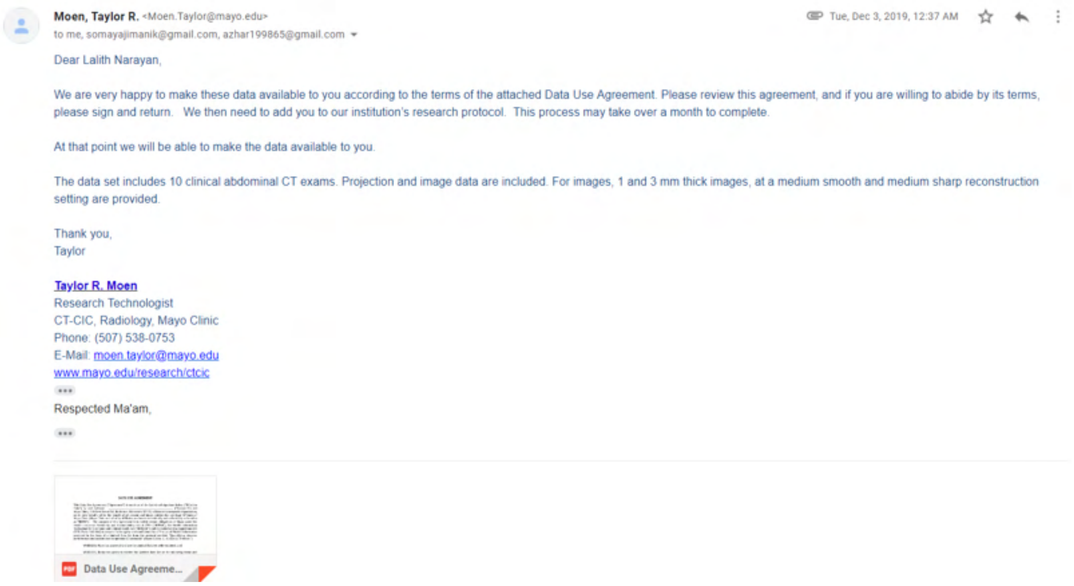
Cynthia

Cynthia H. McCollough, PhD, FAAPM, FACR, FAIMBE
Brooks-Hollern Professor
Professor of Medical Physics and Biomedical Engineering
Director, CT Clinical Innovation Center and X-ray Imaging Core
Department of Radiology

Figure 4.2: Email Requesting the Authorities for Dataset

4.1.2 Dataset Acquisition

The dataset is made publicly avaialble by the American Association of Physicist in Medicine and Mayo Clinic. We requested the respective authorites for the dataset, and signed a Non-Disclosure Agreement regarding the dataset. After a time frame of two months, we got access to the dataset on February 13, 2020. We have attached copies of the mail, in figures 9.1 and 9.2 signalling the evidence for the same.



Moen, Taylor R. <Moen.Taylor@mayo.edu>
to me, somayajimanik@gmail.com, azhar199865@gmail.com

Tue, Dec 3, 2019, 12:37 AM

Dear Lalith Narayan,

We are very happy to make these data available to you according to the terms of the attached Data Use Agreement. Please review this agreement, and if you are willing to abide by its terms, please sign and return. We then need to add you to our institution's research protocol. This process may take over a month to complete.

At that point we will be able to make the data available to you.

The data set includes 10 clinical abdominal CT exams. Projection and image data are included. For images, 1 and 3 mm thick images, at a medium smooth and medium sharp reconstruction setting are provided.

Thank you,
Taylor

Taylor R. Moen
Research Technologist
CT-CIC, Radiology, Mayo Clinic
Phone: (507) 538-0753
E-Mail: moen.taylor@mayo.edu
www.mayo.edu/research/ctcic

Respected Ma'am,

Figure 4.3: Response Email from Dr.Moen Taylor, Mayo Clinic

Chapter 5

Dataset Pre-processing and Data Format

5.1 DICOM

DICOM stands for Digital Imaging and Communications in Medicine. It is a medical standard for the communication and management of medical imaging information. The DICOM standard covers the aspect of storing the medical data and also the protocols to be applied for communicating different devices such as TCP/IP and services in the imaging workflow. DICOM was developed in the year 1993 by the initiative of American College of Radiology (ACR) and the National Electrical Manufacturers Association (NEMA). It is referred to as "DICOM 3.0", as it is an evolution of the previous ACR-NEMA 2.0 standard. The National Electrical Manufacturers Association (NEMA) holds the copyright for the current standard .

The main intention and motivation for developing DICOM standard is to allow cross-vendor inter-operability among devices and information systems dealing with digital medical images. For Example vendor 1 should be able to send a report to a digital archive of vendor 2, or a diagnostic workstation belonging to vendor 3.

With its success, DICOM has become the de-facto standard in medical imaging. Today almost every digital imaging systems of all major vendors(including acquisition devices, diagnostic workstations, archives, servers, medical printers, etc.) use the DICOM standard . Also, DICOM has been widely accepted and adopted by medical institutions, including public and private hospitals, diagnostic centres and analysis laboratories of different sizes.

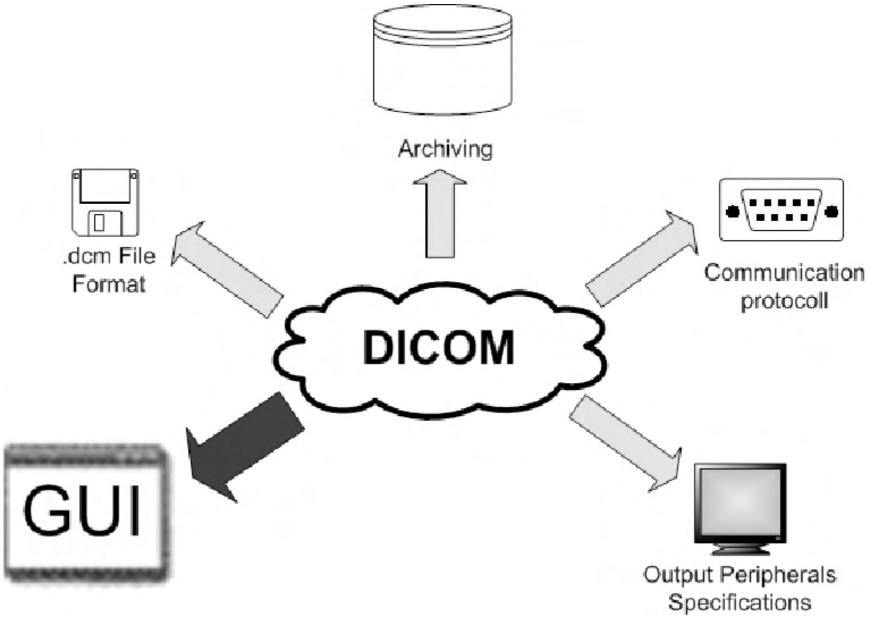


Figure 5.1: Illustration of DICOM Standard

The advantages of DICOM are:

1. It makes the medical data interoperable.
2. Helps in efficient and hassle-free integration between various devices from different manufactures.
3. It has large developing community to meet the evolving technology of medical imaging.
4. Is free to use and many libraries are developed in various programming languages for research.

DICOM files contain metadata which provide information about the image data, such as the size, dimensions, bit depth, modality used to create the data, and equipment settings while capturing the image, patient id ,name etc.

5.2 How DICOM images are taken

The dataset we use contains the helical CT images. Earlier the CT Scanners used to image one slice at a time before moving to next slice. So the disadvantage associated with this is that the thickness of the slice would completely depend on the CT Detector. Because the scanners collimate with the beam width of the detectors, therefore slice thickness will be also be equal to collimator width.

The current technology uses the helical or spiral scanners. Here the scanner constantly rotates around the patient and moves. It goes around the patient in a helical manner.

Thus since two scanners are moving no two projections correspond to a single slice at some z-coordinate. The scanner constructs a slice after completing one 360 degree rotation around the body. Modern helical CT scanners interpolates the data projections 180 degree apart. So a photon suffers the same attenuation as it passes through the same tissue. This interpolation is done to convert the helical path to the transverse slices. This also explains the concept of slice broadening because each point reflects data from two z positions, separated by the amount the table moves as the radiation beam turns 180 degrees. As before, the thickness of tissue included in a single projection is the collimator width. The relationship between speed of rotation of the CT scanner and speed of table movement is referred to as pitch.

$$Pitch = \frac{Tablemovementin360degrees}{Collimatorwidth} \tag{5.1}$$

Thus if the pitch =1 then for 360 degree rotation of the scanner around the body the collimator starts adjacent to where it started 360 degree before with no gap. If pitch is less than 1 then there will be overlap of the helices. I pitch is greater than one then the gaps will be more creating the complexities in construction of the slices. Generally the pitch is kept between one and two. The advantage of helical CT scanner is that it is much faster than the step-and-shoot scanner since the entire scan is acquired in one continuous motion. Increasing pitch increases the speed of the scan. But it comes with the disadvantage of loosing of information between the gap which causes introduction of noise and artifacts while reconstructing. The decreased pitch increases the quality of image but time taken is more and the patient will have to be exposed to radiation for longer time. Below are the two snapshots of the path of helical CT scanner with different pitch.

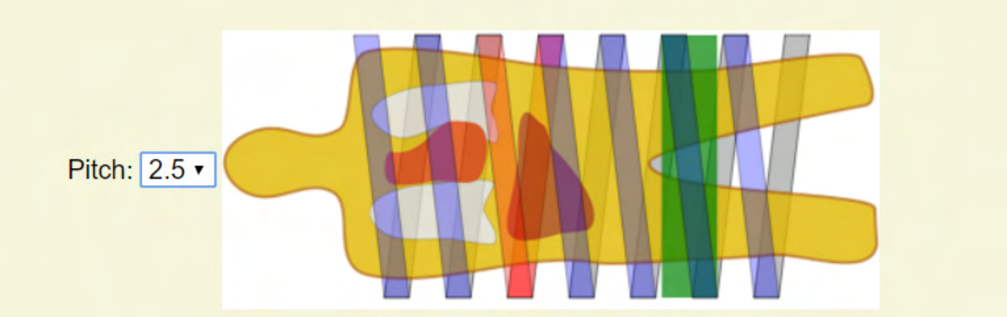


Figure 5.2: CT scanning path for pitch=2.5

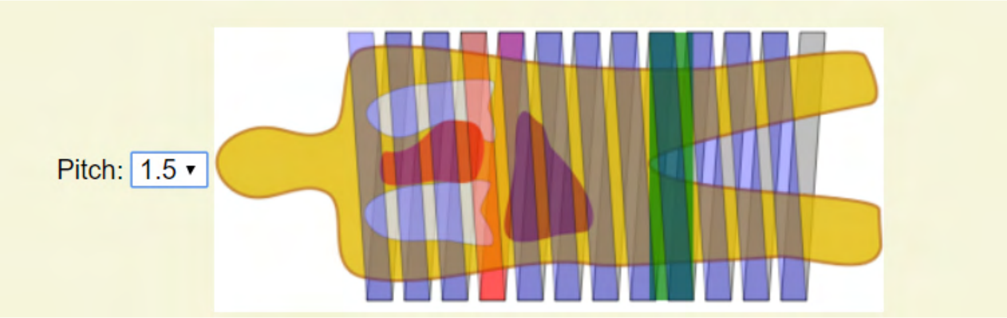


Figure 5.3: CT scanning path for pitch=1.5

5.3 Data-Format of DICOM Files and Pre-processing of the Dataset

A DICOM element or attribute consists of the following important properties:

1. A tag which identifies the DICOM element in the format. (XXXX,XXXX) hexadecimal system . This tag can be further divided into DICOM group number and DICOM element number.
2. A DICOM value which describes the data type and the format of the attribute value.

So working with these files can be challenging, especially with lot of metadata containing the details of heterogeneous environments, equipments where the CT Scan is taken. Lot of preprocessing needs to be done in order to bring all the images to some standard form so that we can send it to our model for processing. The pre-processing before feeding it to our machine learning model involves:

1. Loading the DICOM files
2. Adding missing meta-data
3. Converting the pixel values to Hounsfield Units(HU)
4. Re-sampling to bring dimensions of all the images to same value
5. Normalisation

As we all know that python is the most prefered language for the machine learning applications because of its rich libraries, so we plan to do the pre-processing part in python itself. Fortunately we have a dedicated python library “dicom”.

5.3.1 Loading the files

DICOM is de-facto medical standard for medical image applications. These files contain lot of meta-data such as pixel size of the image captured by equipment. This pixel sizes vary from machine to machine and from scan to scan. Another metadata which is important to be worried about is the distance between the samples which hurts the ML model performance. So this step we load a scan corresponding to one patient which consists of multiple slices, However the data set we use has a missing field about the pixel size in z-direction.

5.3.2 Adding missing meta-data

As discussed earlier we have a missing meta data about the pixel size in z-direction. Luckily we can infer this by other metadata.

```
slice_thickness = np.abs(slices[0].ImagePositionPatient[2]
- slices[1].ImagePositionPatient[2])
```

or

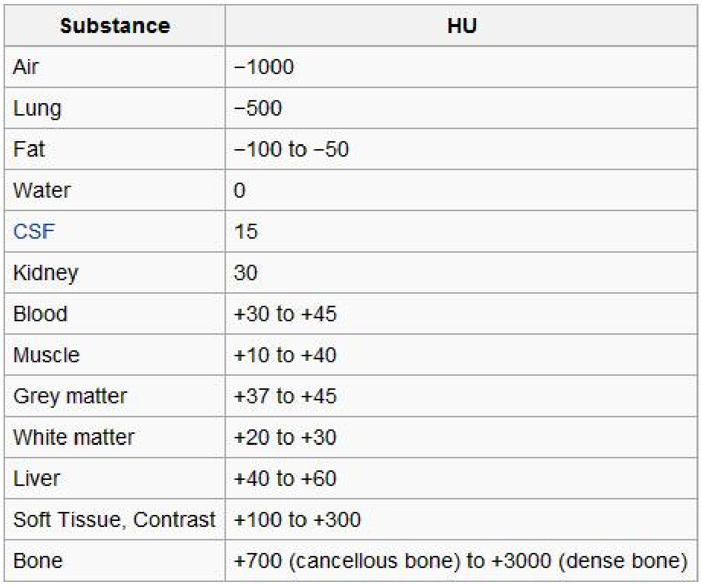
```
slice_thickness = np.abs(slices[0].SliceLocation
- slices[1].SliceLocation)
```

where SliceLocation and ImagePositionPatient are the two metadatas we use to get slice thickness

5.3.3 Converting the Converting the pixel values to Hounsfield Units(HU)

Since each image is taken in different environment or lighting conditions or disturbances in radiation intensities so we need to bring it to a standard form. The unit of measurement of CT Scans is Hounsfield Unit(HU), which is measure of radio-intensity. Below is the table which describes the Hounsfield values of

different substances and body organs.



Substance	HU
Air	−1000
Lung	−500
Fat	−100 to −50
Water	0
CSF	15
Kidney	30
Blood	+30 to +45
Muscle	+10 to +40
Grey matter	+37 to +45
White matter	+20 to +30
Liver	+40 to +60
Soft Tissue, Contrast	+100 to +300
Bone	+700 (cancellous bone) to +3000 (dense bone)

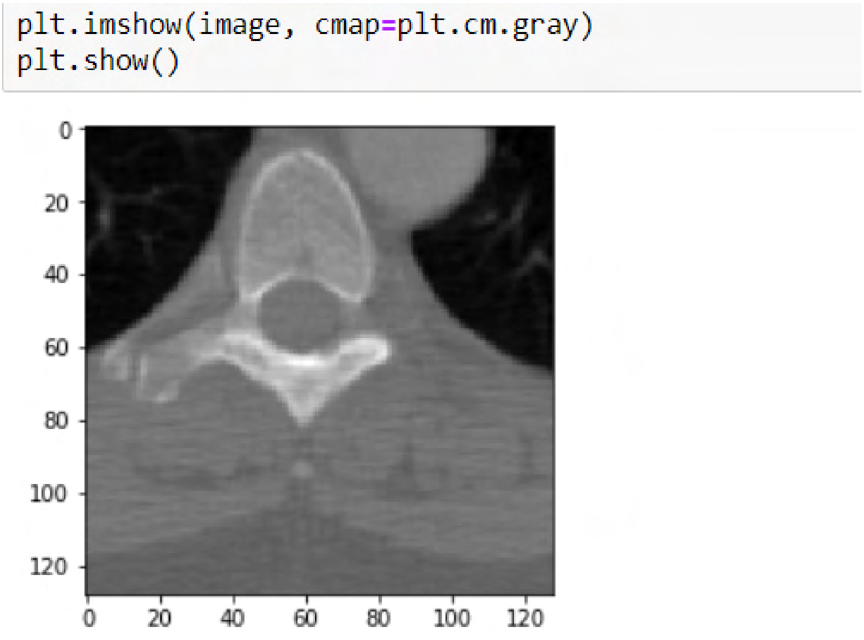
Figure 5.4: Hounsfield Table for different substances

So we fix this problem by converting to Hounsfield value. Some scanners have the cylindrical bounds but the output image is square so the pixels which falls outside the bound of square will get the fixed value -2000.

1. As the first step we set these values to zero(Which is Hounsfield value of air).

2. Next step we multiply each pixel with rescale slope and adding the rescale intercept which are available in the DICOM files the meta data.

```
image[slice_number] = RescaleSlope *
image[slice_number].astype(np.float64) + RescaleIntercept
```



```
plt.imshow(image, cmap=plt.cm.gray)
plt.show()
```

Figure 5.5: The output after preprocessing

5.3.4 Resampling

For a particular scan the slice spacing be 2.5 and for some other scans the pixel spacing may be 1.5. This leads to a problem for automatic analysis. So a common practice is to use re-sampling the whole dataset to same resolution.

5.3.5 Normalisation and patch extraction

Normalisation of the image is done and patch extraction is done on image because DL-based methods need a huge number of samples. This requirement cannot be easily met in practice, especially for clinical imaging. In this study, we propose to use overlapped patches in CT images. This strategy has been found to be effective and efficient, because the perceptual differences of local regions can be detected, and the number of samples are significantly boosted. In our experiments, we extracted patches from LDCT and corresponding NDCT

Chapter 6

Auto-Encoders

RED-CNN

With the risk of high exposure of the radiation to the patient, low dose medical imagery is a trending interest in the medical research field. Low dose exposure means low x-ray flux which results in the noisy CT Image i.e less signal to noise ratio. There are lot of other techniques to use the low dose ct scan and achieve results as comparable to high dose ct scan . like: (a) sinogram domain filtration, (2) iterative reconstruction, and (3) image processing. Deep learning

is the other way to handle this above problem. With the evolution of deep learning algorithms and their ability to handle the complex problem, in this project we try to explore some of the algorithms to solve the above problem. The proposed problem is complex as the task is to remove the noise without effecting the structural properties and details. So we require a generative model . The first algorithm we are going to explore is Red-CNN.

6.1 Introduction

Red-CNN the short name for Residual Convolutional Neural Networks[13]. This is a simple CNN with residual connections . As the number of layers increase, the number of convolutions increase and chance of the structural details diminishing increase. So we need to incorporate the de-convolutional network.
Red-CNN is a auto-encoder decoder network. Auto encoder and decoder network is artificial networks which learns the data distribution in an unsupervised way. The aim of an auto encoder is to encode the input data by doing dimensionality reduction. On the other side decoder takes the input from the encoded output of encoder and learns to reconstruct the Input image back with some extra features. Thus acting as generative model. Several varients of this networks exsists such as Sparse, Denoising, Constructive autoencoders. The basic structure of the auto-encoder and decoder is shown below

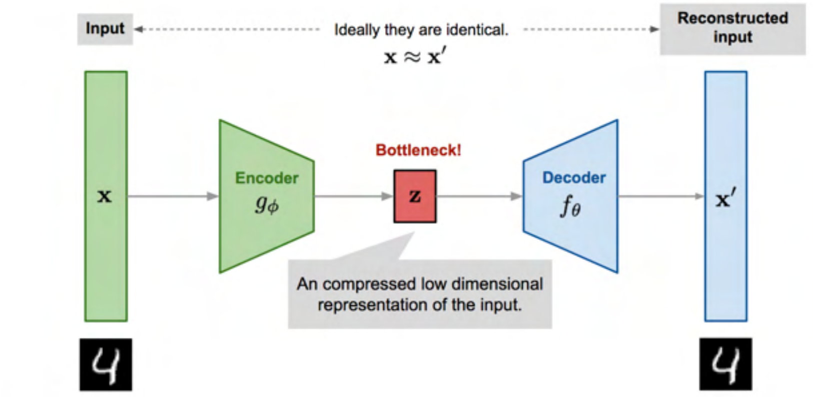

Input
Ideally they are identical.
$\mathbf{x} \approx \mathbf{x}'$
Reconstructed input
x
Encoder
g_ϕ
Bottleneck!
z
Decoder
f_θ
$\mathbf{x}'$
An compressed low dimensional representation of the input.

Figure 6.1: An image of a Auto-Encoder

6.2 Structure of RED-CNN

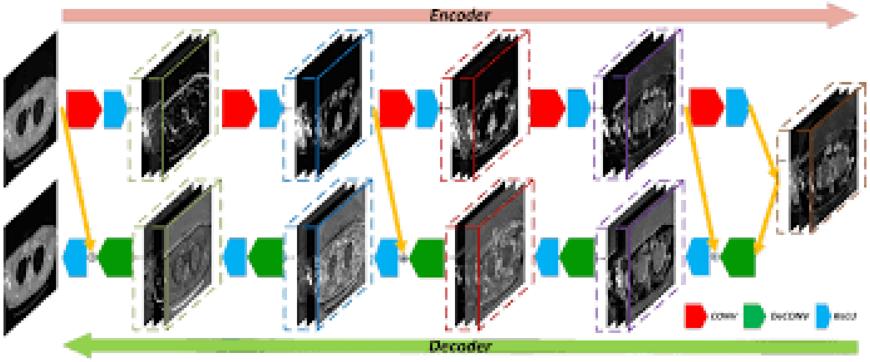


Figure 6.2: Structure of the RED-CNN we use

This network as discussed earlier consists of two sub-networks :

6.2.1 Encoder Network

This consists of chain of fully connected convolutional layers. These layers help in reducing the noise and the artifacts. It is common practice to use the pooling layers in CNN's in order to bring the output to desired size. But it has been found that using a pooling layer suppresses the structural details as well and there is no inverse layer of pooling which we can use in decoder network. So we discard the use of pooling layer. As a result we just have a convolutional layers and ReLU units attached to each of them. So we formulate each layer as follows:

$$C_c^i(y_i) = ReLU(W_i^` \otimes b_i^`)i = 0, 1, 2...N \tag{6.1}$$

Where N is the number of convolutional layers. W and b represents weights and biases respectively

6.2.2 Decoder Network

Even though we have discarded the pooling operation in encoder network, the convolution done there diminishes the high level details in the Ct image. So in order to bring back the structural details we use the decoder network. The decode network consists of series of deconvolutional layers stacked together. Deconvolution is the inverse of the convolution. This is also called as Transposed or fractionally strided convolutions. This is a misnomer in some sense as the deconvolution layer does the convolution action itself.

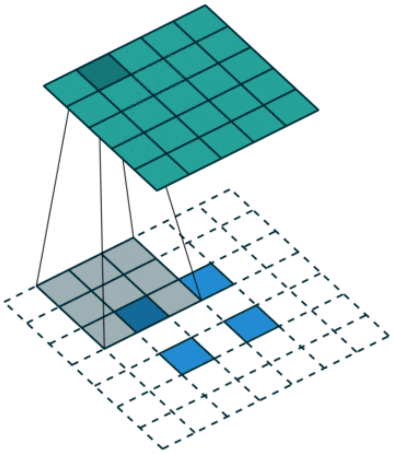


Figure 6.3: Deconvolution action

Consider the image shown, We demonstrate how a 2x2 is converted to 5x5 using deconvolution. We have set the kernel size to be 3x3. The blue pixels correspond to the input image. We pad zeros to the 2x2 input matrix to make it 7x7 matrix and then perform convolution to create a 5x5. This process may not be the exact mathematical inverse but it helps a lot in encoder and decoder type of networks We stack these deconvolutional layers with a Relu unit attached to each of the layer to bring back the structural details. In order to ensure the same input and output size we need to use the same Kernel size and the whole network has to be symmetrical at the middle i.e the encoder network and decoder network has to be same and the dataflow from convolutional and de-convolutional layer has to follow FILO(First In Last Out). So the last convolutional layer should correspond to the first de-covolutional layer. The de-convolutional layer is formulated as:

$$D_d^i(y_i) = ReLU(W_i^{`} \otimes b_i^{`})i = 0, 1, 2...N \tag{6.2}$$

Where, N is the number of de-convolutional layers. W and b represents weights and biases respectively.

6.2.3 Residual Mapping

We First talk about the advantages of the depth of the network and the problems associated with it.Then we explain why residual mapping is used.

Advantages and Disadvantages of the Depth

We know that more shallow is the network ,lesser will be its characteristic of generalisation. Shallow network is good at memorisation. As the number of layers increase at various level of abstraction different features is learnt for example first layer may extract features like average value second layer may extract edges or foreground. So as the layers increases, more complex problem can be generalised[16]. . Multiple layers are much better at generalizing because they learn all the intermediate features between the raw data and the high-level classification. Keeping the advantages in mind we also encounter disadvantages by increasing number of layers. As the size of network increases, especially in case of this network as we have two sub networks. The problem will become more ill-posed and complex. The accumulated loss can prove to be insufficient for back-propagation and since there are lot of Relu units there might be high chance of network suffering Vanishing-Gradients. In order to avoid that we use Residual mapping. Vanishing Gradient Problem was a major problem 10 years back when the people used to add more and more layers in order to meet the complexity of the deep learning problem. But as the number of layers increased, the accumulated loss at the end of the network became insufficient in order to train earlier layers. The gradients of error with respect to weights tends to get smaller and smaller as the error back propagates. It is extremely important for a network to have a well trained earlier layers because the first layers normally extracts the features and the region of interest in a problem. As a result of the network will stop learning or training becomes slower and insufficient. There are many ways to avoid this such as gradient clipping, residual mapping, densely connected network etc The residual mapping works as follows: Let O be output

Image, I be input image, then Residual mapping is denoted by

$$F(I) = O - I \tag{6.3}$$

We let the network to train to fit F(I).Then the output is reconstructed by

$$R(I) = O = F(I) + I \tag{6.4}$$

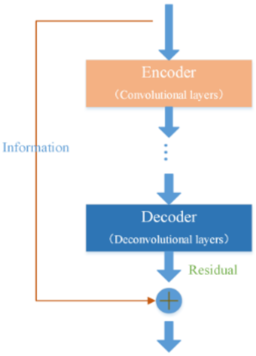


Figure 6.4: Residual mapping

This residual connection do not go through any activation function that reduces the problem occurring due to the derivative, Thus solving vanishing gradient problem. Since aim is to make F(I) as zero as possible such that I = O i.e structural and contrast details can be preserved in the output, which can significantly enhance the LDCT imaging performance. In our project we

use 5 convolutional layers with each layer having Relu unit in Encoder and 5 deconvolutional layers. There exists 3 residual connections between encoder and decoder layers. The residual connections are between

1. Input of encoder and the last layer of decoder.
2. Input of third layer of encoder to the output of third layer in decoder.
3. Input of last layer of encoder to the output of first layer of decoder.

6.3 Training

We have used **pytorch** library to train this model.The network training is optimized using the Adam Optimization algorithm, The optimization procedure is shown in figure 7.10. The batch size is set to 32 and the image patch size is set to 80x80. The parameters for Adam optimizer were set as $\alpha = 1e - 5, \beta_1 =$

$0.5, \beta_2 = 0.9$ and the control parameter λ is chosen as 10. The proposed network is an end-to-end mapping from low-dose CT images to normal-dose images. Once the network is configured, the set of parameters,

$$\Theta = \{W_i, b_i, W_i^{'}, b_i^{'}\} \tag{6.5}$$

of the convolutional and deconvolutional layers needs to be estimated in order to construct the mapping function M. The estimation can be achieved by minimizing the loss $F(D;\Theta)$ between the estimated CT images and the reference NDCT images. Given a set of paired patches $P = (X_1, Y_1), (X_2, Y_2)...(X_N, Y_N)$ where $\{X_i\}$ and $\{Y_i\}$ denote NDCT and LDCT image patches respectively, and K is the total number of training samples. The mean squared error (MSE) is utilized as the loss function:

$$F(D;\Theta) = \frac{1}{N}\sum \|X_i - M(Y_i)\|^2 \tag{6.6}$$

In this study, the loss function was optimized by Adam Optimiser

Chapter 7

Wasserstein-GAN

7.1 Introduction

GANs or Generative Adversarial Neural Networks[3] are a class of generative methods that are used to approximate a true data distribution, This approximation is learned adversarially in the form of a min-max game played between the generator $G(z;\theta_g)$ and discriminator $D(x;\theta_d)$, Where both D and G are differentiable functions represented by convolutional neural networks with parameters θ_d and θ_g respectively. The objective of the generator is to produce realistic images to fool the discriminator and the objective of the discriminator is to tell apart the fake and real images.

Mathematically this means given a true data distribution $P_{data}(x)$ and a prior probability distribution $P_z(z)$ over the noise variable z the GAN framework learns a generator $G(z)$ that approximates the true data distribution $P_{data}(x)$.To learn the parameters θ_d and θ_g the following value function is optimised.

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

Here the Discriminator maximises the value function and the Generator minimises the value function. During the initial stages of training the log term $log(1 - D(G(z))$ does not provide a strong enough error signal to the generator due to which the discriminator learns at a faster rate than the generator leading to a poor generator. To mitigate this problem the generator is set to maximise the following log term $log(D(G(z))$ instead, This provides the generator with sufficient error signal to start learning the optimal parameters θ_g and eventually fool the discriminator.

W-GAN

With many methods being developed to solve the problem of using low-dose CT scan images, there has been a constant problem of scalability. The method-

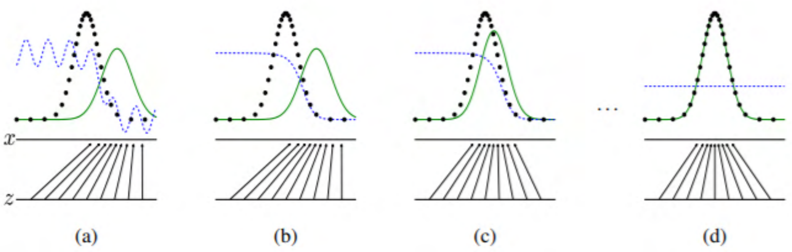


Figure 7.1: Generative adversarial nets are trained by simultaneously updating the discriminative distribution(D, blue, dashed line) in order to discriminate between samples from the original data generating distribution(black, dotted line) P_∞ from that of generative distribution $P_g(G)$(green, solid line). Cases a), b), c) and d) indicate various cases of training.

ologies developed fail at generalising data thus limiting the commercial usage of the same. On the onset of deep learning algorithms, more confidence was infused in their ability to generalise over very large sets of data. The overview looked at so far involves the RED-CNN, which is based on mean-squared error loss. In the upcoming section, we will consider the disadvantage of using MSE loss.

7.2 Disadvantages of MSE Loss

Perceptually, the MSE loss function signifies pixel to pixel comparison of the generated CT image and the normal-dose CT image. The per pixel MSE tends to overlook subtle features associated with the image, and therefore does not take into account the visual features for loss computation. In order to understand the overlooking of various features taking place in a neural network, we need to consider the various steps in a neural network. Let us assume that CT image has a distribution function over a set of manifolds. Understanding the blurring process requires an analysis of the neural networks.

7.2.1 Neural Networks

Visualizing deep neural networks[4] is always challenging, and it remains a black-box majority of the time. On the contrary, visualizing neural networks gives us insights and thus we can use low-dimensional neural networks as they can be visualized easily. With that being said, neural networks go through a process of creating representation. Representation is the new set of data transformed via a layer in the network. With addition of layers, the networks transforms the data and the manifolds to create linearly separable boundaries. Homeomorphisms are instances of topological equivalence. Topological equivalence is the rela-

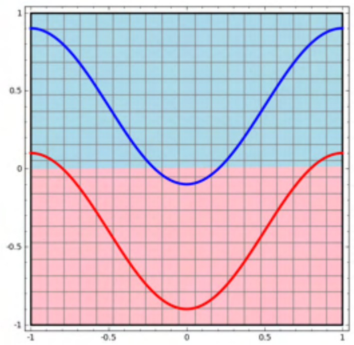


Figure 7.2: Initial Boundary

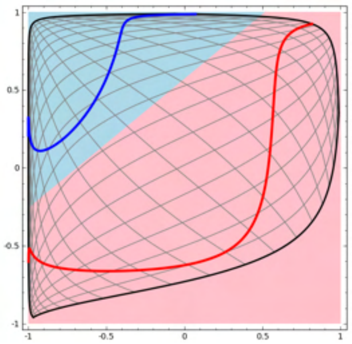


Figure 7.3: Final Boundary Learnt by the Neural Network

tionship between two representations, capable of being transformed into each other by a one-to-one transformation continuous in both directions. Checking for homeomorphisms help us visualize and understand the layers of neural networks. Let us prove that neural network representations are homeomorphisms.

To accomplish the same[4], we consider the various operations performed on the layers of a neural network. The nodes in a neural network during training goes through the following steps:

1. Linear transformation by weight matrix W
2. Translation via the intercept/bias vector b
3. Point-wise application of activation function

Theorem 1. *Layers with N inputs and N outputs are homeomorphisms, if the weight matrix W is non-singular.*

Proof. To prove the above theorem, we consider the various operations on a neural network and prove that these operations are homeomorphisms.

1. Let's assume W is non-singular, that is, it's determinant is non-zero. Then it is a bijective linear function with a linear inverse. Linear functions are continuous. Therefore, The operation of multiplication by weight matrix W is a homeomorphism.
2. Translation operation is a homeomorphism.
3. Activation function are continuous functions in their respective domains, with continuous inverses.Therefore, the operation of activation function is a homeomorphism

□

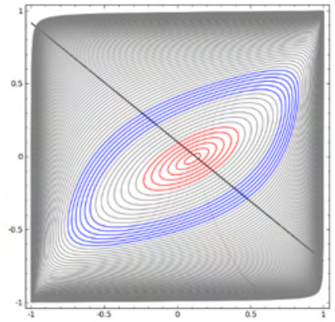


Figure 7.4: Initial Boundary

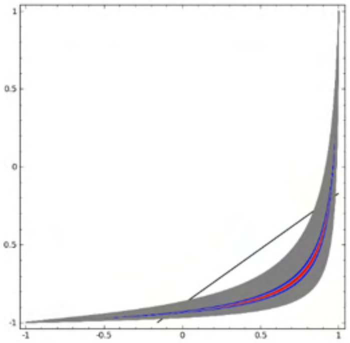


Figure 7.5: Final Boundary Learnt by the Neural Network

Refering to figure 7.4 and figure 7.5, we observe the inital learning boundary and the final learning boundary. The neural network fails to learn the boundary with a 100 percent accuracy. On the contrary, the final model works well for 80% of the time. Therefore, the accuracy may not be a suitable metric to measure the level of generalization of a neural network.

As the number of layers increase, the degree of generalisation increases. This degree of generalisation is irrespective of the distribution of the dataset, and depends on the objective function. Therefore, this explains the failure of MSE loss and the blurrin effects introduced.

Therefore, additional attention needs to be paid at defining a loss function that takes care of these averaging affect of neural networks.

7.3 Wasserstein GAN

Wasserstein GAN[5] also known as W-GAN is an extension of the GAN-framework. W-GAN addresses the fact that GANs during training usually lead to saturation, and therefore this limits the ability of the GAN to learn. The log term present in the GAN's loss function usually drives it to saturation. W-GAN is alternative presented to train the generator model in order to better approximate the distribution of the dataset. In the following sub-sections we cover W-GAN in depth.

7.3.1 Problems with Likelihood Functions

The main problem of concern here is to learn the probability distribution of the low-dose CT images. Usually, learning the probability distribution refers to modelling the probability density function. To attain the same, a parametric family of densities, $(P_\theta; \theta \epsilon R^d)$ is defined, and the one that maximized the likelihood on the dataset is treated as the model.

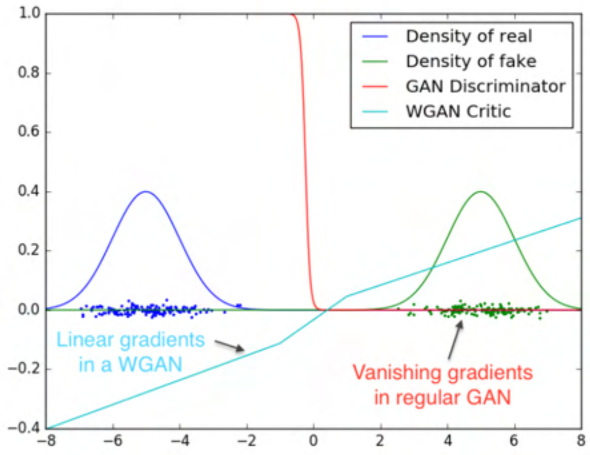


Figure 7.6: Optimal Discriminator and critic when learning to differentiate between two Gaussians. The discriminator of a minmax GAN saturates and resilts in vanishing gradient. WGAN generates clean gradients on all parts of the space.

Let us assume the real distribution to be (P_r) and (P_θ) is the distribution of the parameterized density. We can treat the above problem as minimizing the Kullback-Leibler Divergence $(KL(P_r||P_\theta))$. For the above apporach to work, we need the model density (P_θ) to exist. In cases when we deal with distributions supported by low-dimensional manifolds, it is highly likely that the model manifolds and the true distribution's manifolds have a negligible intersection, and in such a case the KL distance is not clearly defined.

Usually the quick solution us to add noise[6] terms to the model distribution. Therefore, all generative models include a noise component. Usually, the noise component is high-bandwidth gaussian noise. The problem with the addition of noise is that it degrades the quality of the samples. The amount of noise required depends on an application basis and therefore the given approach fails.

The approach that works is to define a random variable Z with a fixed distribution $p(z)$ and pass it through a parametric function $g_\theta : Z \implies X$ that would directly generate samples following a certain distribution P_θ. The parameter θ can be varied to make the distribution as close to the real distribution P_r. The above approach works well in two ways:

- This approach can represent representations of low-dimensional manifolds.
- Ability to easily generate samples is more useful in many applications than having information about the numerical value of the probability density.

Moreover, it aids the algorithms computationally, as generating samples belonging to high-dimensional density can be computationally challenging

The various approaches taken towards generative models are auto-encoders and generative adversarial networks. Auto-encoders focus on the approximate likelihood of the examples, and therefore share the limitations stated earlier with likelihood based objective functions. GANs, on the contrary, offer lenience in the definition of the objective function.

7.3.2 Earth-Mover Distance

The original GAN consists of the generator and the discriminator. The discriminator outputs the probability that its input is fake. The ultimate goal of the training is to approximate the real data distribution P_r with the generated distribution P_g. The cross-entropy is used as the loss function which corresponds to the problem of optimization of Jensen-Shannon (JS) divergence between the two distributions. The JS divergence is a distance metric that signals the distance between the two probability distribution.

Various distance measures[5] were used to model the difference between distribution functions, such as JS divergence, KL-divergence, Total-variation distance. These distance measures fail at many areas such as continuity of loss functions, vanishing gradients problem during training or dependency on probability densities that might not be available.

Therefore, the *Earth-Mover* (EM) distance was introduced which is defined as follows

$$W(P_r, P_g) = \inf_{\gamma \epsilon \pi(P_r, P_g)} E_{(x,y)\sim\gamma}[||x - y||] \tag{7.1}$$

where $\pi(P_r, P_g)$ denotes the set of all joint distributions $\gamma(x, y)$.

A convenient dual for the same is given by

$$W(P_r, P_g) = \frac{1}{K} \sup_{\|f\|_L < K} E_x[f(x)] - E_z[f(G(z))] \tag{7.2}$$

where the supremum is taken over functions where the norm of their respective gradients is always lesser than K.

On an intuitive level the term $\gamma(x, y)$ indicates the amount of "mass" that needs to be transported from x to y in order to transform the distributions P_r into P_g. Earth-Mover distance is the cost of the optimal transport plan.

Considering an example of learning parallel lines, we can prove that EM distance works the best, whereas the other distances fail at the task. Let $Z \sim U[0.1]$ be an uniform distribution. Let P_0 be the distribution of (0,Z) ϵR^2. The figure 7.7 shows that EM distance provides a useful gradient, whereas the JS divergence does not provide the same.

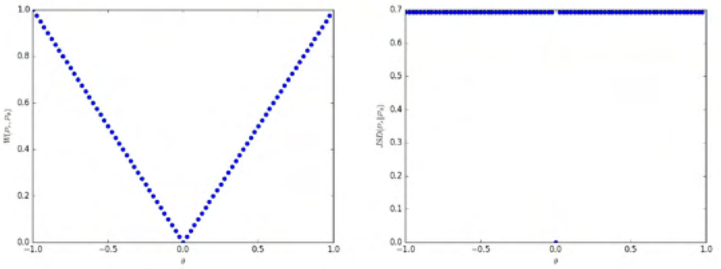


Figure 7.7: These plots show EM distance and the JS divergence on left and right plots respectively. The EM plot is continuous and provides a usable gradient everywhere, whereas JS plot is not continuous and therefore gradient is of no value.

The above example tells us that low dimensional manifolds can be learnt by doing gradient descent on EM distance, but is not possible with other distances.

7.3.3 Gradient Penalty

The original WGAN paper used weight clipping to limit the discriminator weights to a specified range. This again posed problems during training. The training was unstable in cases of large datasets, and therefore the above approach was modified to soft clipping. This term was coined as gradient penalty[7], which is expressed as

$$(\|\nabla_{\hat{x}} D(\hat{x})\| - 1)^2 \tag{7.3}$$

where we take the gradient at a randomly weighted average of the real and the generated samples, $\hat{x}$. $\hat{x} = \epsilon X + (1 - \epsilon)G(z)$ where ϵ is selected randomly between 0 and 1. Therefore, the discriminator loss is modified to the following equation

$$L_D = D(x) - D(G(z)) + \gamma(\|\nabla_{\hat{x}} D(\hat{x})\| - 1)^2 \tag{7.4}$$

An intuitive explanation for the above loss function is as follows: The discriminator learns to output very large values for fake samples and extremely small values for real samples, thereby increasing the confidence of prediction. The gradient penalty term monitors and stops the weights from increasing exponentially.

The WGAN solves the following minmax problem,

$$\min_G \max_D L_{WGAN}(D, G) = -E_x[D(x)] + E_z[D(G(z))] + \lambda E_{\hat{x}}[(\|\nabla_{\hat{x}} D(\hat{x})\| - 1)^2] \tag{7.5}$$

where the first two terms indicate the Wasserstein distance estimation and the last term indicates the gradient penalty term.

7.4 Perceptual Loss

The wasserstein distance aids the process of learning lower dimensional distributions as well. On the contrary, if we use root mean square error as the loss function, it induces blurring as we have seen in the earlier section. Therefore, perceptual loss is introduced. The reason behind using the above loss function is two-fold. The first and foremost reason is an intuitive one. When we compare two images, we are comparing them on a basis of the features. It is not a pixel to pixel comparison. Our visual system extract features from images, and then compares the two. Therefore, a loss function which compares images should consider features as a basis for comparison. Hence, VGG-19 network, which is a pretrained model developed by Google is used for feature extraction. The denoised output is compared against the normal dose output in the feature space. Mathematically speaking, CT images are not uniformly distributed in a higher dimensional Euclidean space. They are more likely to localize in a lower dimensional manifold. With MSE, these lower dimensional manifolds are generalized and therefore discarded. Moreover, we are not measuring the intrinsic similarity in the case of MSE loss. In the case of perceptual loss, we are comparing them by projecting them onto a manifold and then calculating the geodesic distance between them. Therefore, perceptual loss aids in producing better results with sharper details.

The generator in the WGAN network learns the distribution and the transformation between the high noise distribution to low noise distribution. The perceptual loss is added to keep image details intact. Using a MSE error, it tries to minimize the pixel wise error between a NDCT image patch x and denoised patch G(z).

$$L_{MSE}(G) = E_{(x,z)}[\frac{1}{N^2}\|G(z) - x\|_F^2] \tag{7.6}$$

where $\|.\|_F$ is the Frobenius norm. The perceptual loss is mathematically defined as follows

$$L_{perceptual}(G) = E_{(x,z)}[\frac{1}{whd}\|\phi(G(z)) - \phi(x)\|_F^2] \tag{7.7}$$

where ϕ is the feature extractor and d,w and h represent the depth, width and height of the feature space, respectively. In our case, $\phi()$ is the VGG-19 model. Therefore, the modified equation is

$$L_{VGG}(G) = E_{(x,z)}[\frac{1}{whd}\|VGG(G(z)) - VGG(x)\|_F^2] \tag{7.8}$$

7.5 Feature Extractor

We use a pre-trained feature extractor VGG-19. State of the art model weights pre-trained on Imagenet dataset are available via various model zoos. These pre-trained models are used for various tasks like feature extraction, image classification and transfer learning. The architecture is displayed in figure 7.8.

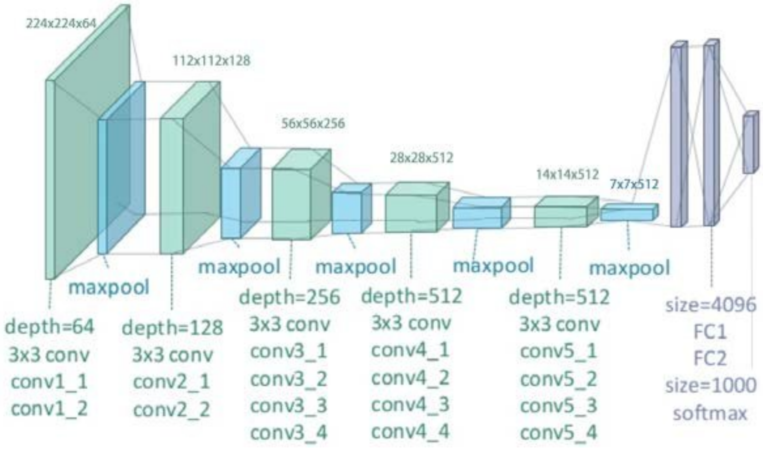


Figure 7.8: VGG-19 Network Architecture

There are various options available when it comes to choosing network architectures. While some offer accuracy, others offer reasonable accuracy with very less hardware requirements. We use VGG-19 as it handles the trade-off well and is the best of the two worlds.

7.6 Overall Loss function

Combining equations (7.5) and (7.7), we can express our overall loss function as

$$\min_G \max_D L_{WGAN}(D, G) + \lambda_1 L_{VGG}(G) \tag{7.9}$$

where λ_1 is the control parameter to control the trade-off between WGAN adversarial loss and the perceptual loss.

7.7 Network Structure and Implementation

The network consists of three parts. They are the generator, the discriminator and the feature extractor. We explore each of these parts in detail in the following sections.

7.7.1 Generator Network

The generator network is a convolutional neural network(CNN), comprising of 8 layers that perform the convolution operation. The inout to the generator is the low dose CT scan image. 3x3 kernels are used in each layer to perform the convolution operation. 3x3 is a standard followed in the deep learning

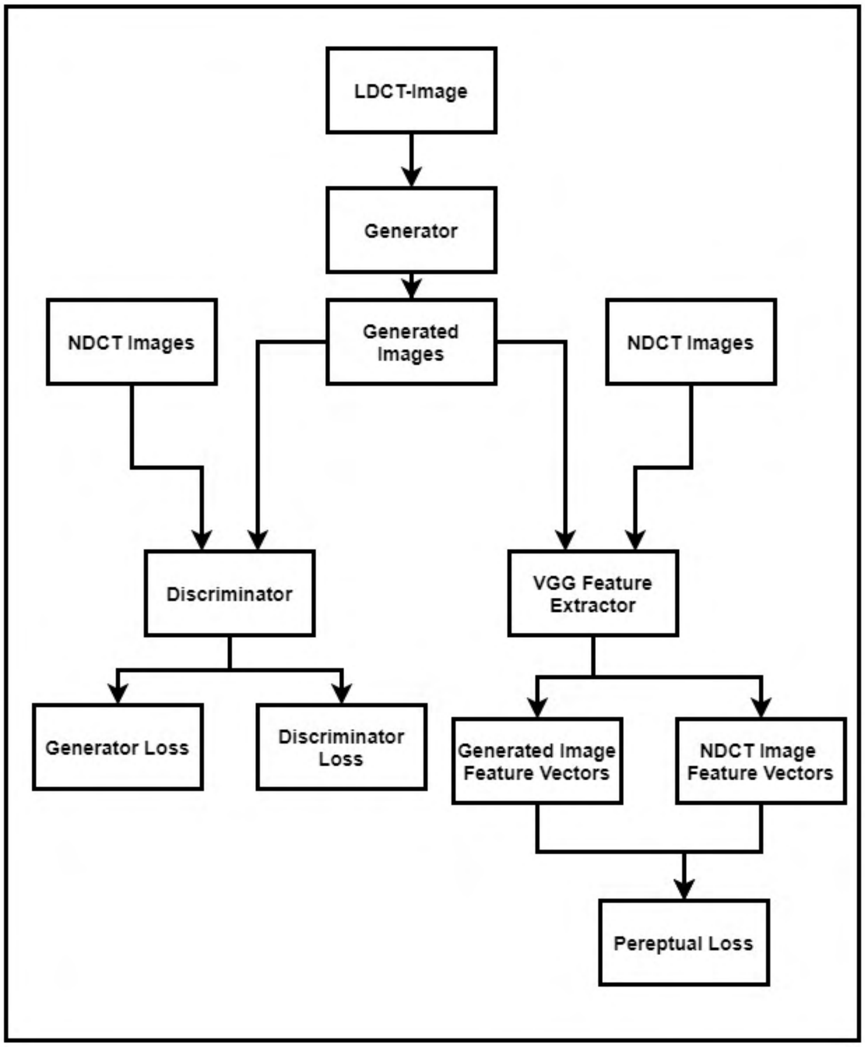


Figure 7.9: WGAN Network Architecture

community. The first 7 layers have 32 filters each, and the last layer has only one filer that generates the feature map. This feature map is the output of the generator, that is fed to both the discriminator as well as the feature extractor.

7.7.2 Discriminator Network

The second part of the network comprises of 6 convolutional layers stacked together to form another convolutional neural network. The discriminator has an increasing number of filters in it. The first two layers comprise of 64 filters, the next two have 128 and the final two have 256 filters in them. After the convolutional layers, there are two fully connected layers with 1024 neurons in the first layer and 1 neuron in the second layer. The discrimator gives the probability that the input image is fake.

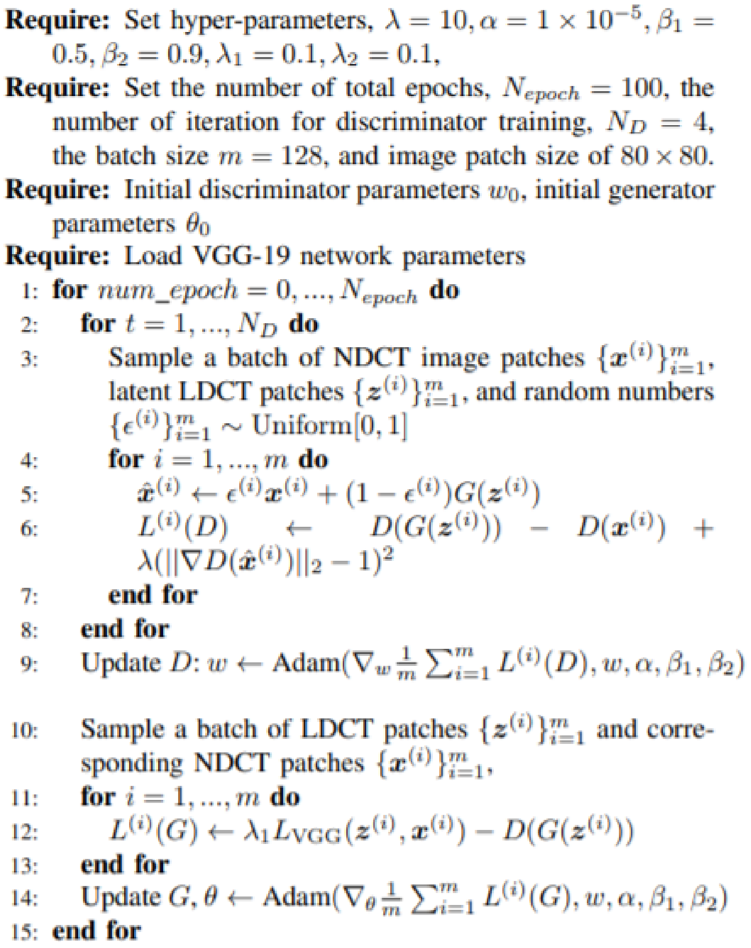
7.7.3 Feature Extractor

The final part of the network is the feature extractor that we have discussed in an earlier section. The inputs to the feature extractor are the generated image G(z) and the normal dose CT image.

7.7.4 Network Training

The network training is optimized using the Adam Optimization algorithm, The optimization procedure is shown in figure 7.10. The batch size is set to 32 and the image patch size is set to 80x80. The parameters for Adam optimizer were set as $\alpha = 1e - 5, \beta_1 = 0.5, \beta_2 = 0.9$ and the control parameter λ is chosen as 10. We implemented the code in TensorFlow[8] and used the inbuilt modules for Adam optimizer[9].

We used a Intel i7-9th Gen CPU combined with Nvidia RTX 2060 GPU. The training took around 3 hours for 9000 iterations of training the samples.



```
Require: Set hyper-parameters, $\lambda = 10, \alpha = 1 \times 10^{-5}, \beta_1 = 0.5, \beta_2 = 0.9, \lambda_1 = 0.1, \lambda_2 = 0.1,$
Require: Set the number of total epochs, $N_{epoch} = 100$, the number of iteration for discriminator training, $N_D = 4$, the batch size $m = 128$, and image patch size of $80 \times 80$.
Require: Initial discriminator parameters $w_0$, initial generator parameters $\theta_0$
Require: Load VGG-19 network parameters
for $num\_epoch = 0, ..., N_{epoch}$ do
    for $t = 1, ..., N_D$ do
        Sample a batch of NDCT image patches $\{\boldsymbol{x}^{(i)}\}_{i=1}^m$, latent LDCT patches $\{\boldsymbol{z}^{(i)}\}_{i=1}^m$, and random numbers $\{\epsilon^{(i)}\}_{i=1}^m \sim \text{Uniform}[0, 1]$
        for $i = 1, ..., m$ do
            $\hat{\boldsymbol{x}}^{(i)} \leftarrow \epsilon^{(i)}\boldsymbol{x}^{(i)} + (1 - \epsilon^{(i)})G(\boldsymbol{z}^{(i)})$
            $L^{(i)}(D) \leftarrow D(G(\boldsymbol{z}^{(i)})) - D(\boldsymbol{x}^{(i)}) + \lambda(||\nabla D(\hat{\boldsymbol{x}}^{(i)})||_2 - 1)^2$
        end for
    end for
    Update $D$: $w \leftarrow \text{Adam}(\nabla_w \frac{1}{m}\sum_{i=1}^m L^{(i)}(D), w, \alpha, \beta_1, \beta_2)$

    Sample a batch of LDCT patches $\{\boldsymbol{z}^{(i)}\}_{i=1}^m$ and corresponding NDCT patches $\{\boldsymbol{x}^{(i)}\}_{i=1}^m$,
    for $i = 1, ..., m$ do
        $L^{(i)}(G) \leftarrow \lambda_1 L_{\text{VGG}}(\boldsymbol{z}^{(i)}, \boldsymbol{x}^{(i)}) - D(G(\boldsymbol{z}^{(i)}))$
    end for
    Update $G, \theta \leftarrow \text{Adam}(\nabla_\theta \frac{1}{m}\sum_{i=1}^m L^{(i)}(G), w, \alpha, \beta_1, \beta_2)$
end for
```

Figure 7.10: Optimization Procedure for WGAN with VGG-19 Network

Chapter 8

Pix2Pix GAN

8.1 Introduction

The Pix2Pix GAN[10] was proposed as the general purpose solution to solve the problem of image to image translation, Apart from learning a functional mapping from input image to output image, The pix2pix GAN also alleviates the problem of carefully designing a loss function for a specific image-image translation problem and requires only the high level specification of the problem in terms of the input and output image. The method has been used to solve a large number of image to image translation tasks like generating images from sketches of objects, automatic generation of segmentation maps from images and many more.In this work we aim to translate a LDCT image to NDCT image.

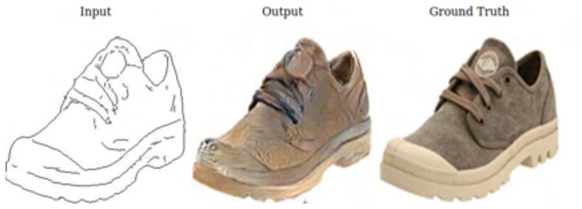


Figure 8.1: pix2pix GAN used to generate image from a sketch

8.2 Objective Function

The pix2pix GAN makes use of the conditional GAN (cGAN)framework. In a cGAN[11] the output of the generator is not only conditioned on the random noise z but also the input image x. The Discriminator is also a function of the input image x. The objective of the cGAN is a modified version of the original GAN[3] objective and is given by the below equation

$$L_{cGAN}(D,G) = E_{x,y}[log(D(x,y)] + E_{x,z}[log(1 - D(x,G(x,z))]$$

Here the Discriminator maximises the Loss while the Generator minimises the Loss. x is the LDCT input image and y the desired NDCT output image. In addition to the cGAN loss term a l1 loss term is introduced to train the generator to produce images as close to the input. The l1 loss is given by the below equation.

$$L_{l1} = E_{x,y,z}[\|y - G(x,z)\|_{l1}]$$

The final Objective given below is optimised using the adam optimizer.

$$L_{total} = L_{cGAN} + L_{l1}$$

8.3 Architecture

The following sections discuss the architecture of the generator and discriminator and the motivation for these design choices.

8.3.1 Generator

The generator is mainly composed of convolutional, batch normalisation modules and leaky ReLU activations. Previous approaches to model the generator include using convolutional neural networks with zero padding to retain the original size of the input image and Encoder Decoder networks like the ones discussed in chapter 6 but without the residual connections. The Generator in pix2pix GAN[10] also uses an encoder-decoder in the form of the U-net[12] architecture which introduces skip connections in the existing encoder-decoder architecture. Skip connections serve the purpose of shuttling low level information from the encoder layer to the decoder layer as a lot of low level features are the same in the encoder and decoder.

Having skip connections provides a shortcut path for relevant information to flow to the decoder without passing through the bottle neck of the encoder-decoder structure. The skip connection is implemented by the simple operation of concatenating features of the ith layer to the (n-i)ith layer in the network or the i^{th} layer of the decoder. The filter size for both the convolutional and deconvolutional layers is set to 4x4. There are 7 convolutional blocks and 7 deconvolutional blocks in the network architecture. The slope of the leaky ReLU is set to 0.2.

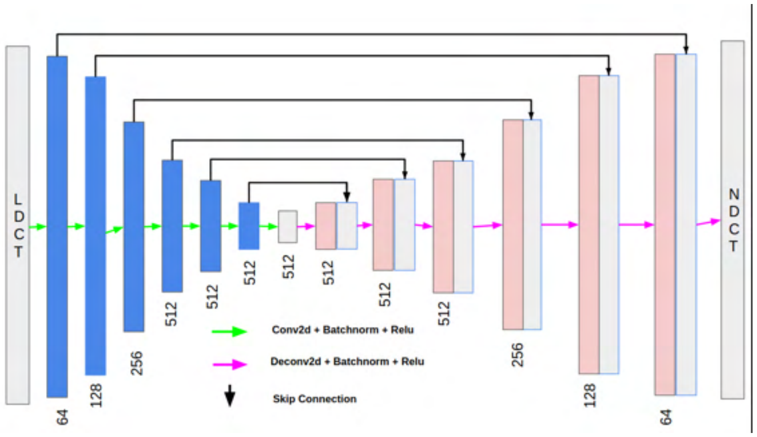


Figure 8.2: Architecture of the pix2pix GAN generator. The blocks represent the feature volume after convolution, the numbers below the blocks represent the channel width.

8.3.2 Discriminator

The Discriminator used in pix2pix GAN[10] is named a patchGAN network. The patch GAN network takes in a NxN patchs of the input image and and outputs a pixel map with each pixel predicting whether the NxN patch of the input image is fake or real instead of predicting the whole image is fake or real. The value of N is dependent on the discriminaror architecture and turns out to be 70, depending on the architecture N can be set to 1 each pixel of the input image is then either predicted to be fake or real and hence the name pixelGAN. Another way of interpreting the PatchGAN is to think of it as fucntion that maps from 256x256 image to an NxN array of outputs X, where each X_{ij} signifies whether the patch ij in the image is real or fake. Which is patch ij in the input?

Well, output X_{ij} is just a neuron in the output layer of the convnet, and we can trace back its receptive field to see which input pixels it is sensitive to. In the pix2pixGAN architecture, the receptive fields of the discriminator turn out to be 70x70 patches in the input image. This is all mathematically equivalent to if we had manually chopped up the image into 70x70 overlapping patches, run a regular discriminator over each patch, and averaged the results.

The discriminator is provided with both with a source/input image and the target image and must determine whether the target is a plausible transformation of the source/input image. Penalizing the generated image on the pixel or patch level results in generating more crisp images as the high frequency components of the image are captured. The PatchGAN configuration is defined using a shorthand notation as: C64-C128-C256-C512, where C refers to a block of Convolution-BatchNorm-LeakyReLU layers and the number indicates the number of filters. Batch normalization is not used in the first layer. The kernel size is fixed at 4×4 and a stride of 2×2 is used on all but the last 2 layers of the model. The slope of the LeakyReLU is set to 0.2, and a sigmoid activation function is used in the output layer.

8.4 Training Details

The training of both the generator and discriminator is carried out by the adam[9] optimizer, The generator indirectly is trained through the discriminator loss. the Generator and discriminator are trained alternatively for the same number of iterations. The model was trained for a total of 30 epochs with batchsize 1 on a google colab Tesla K80 GPU.

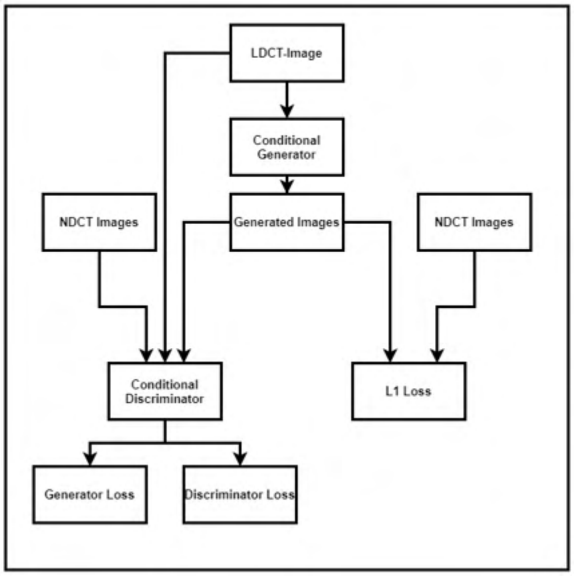


Figure 8.3: Block diagram of pix2pix GAN

Chapter 9

Results

The experiments performed lead us to a series of results, that we studied both quantitatively, and qualitatively.

9.1 Quantative Results

For quantitative analysis, we consider various parameters like Peak Signal to Noise Ratio(PSNR), Structural Similarity Index(SSIM), and Root Mean-Square Error(RMSE). Lets first try to understand what does these parameters mean

- **PSNR** -> PSNR stands for Peak Signal To Noise Ratio. It defines the ratio of maximum possible power of the signal to the maximum power of the distorting noise signal. A higher value of PSNR indicates that the denoising is of higher quality.

- **SSIM** -> SSIM stands for Structural Similarity Index Metric. This quantity describes how perceptually similar any two given images are. It considers the perceived change in the structural information caused by the degradation of the image.

- **RMSE** -> RMSE stands for Root Mean Squared Error. It represents the square root of the second moment of the difference between the two images which are required to be compared. Although not the best metric to use RMSE gives us a baseline on the quality of images produced by the model

Overview of Results			
Model Name	Average PSNR	Average SSIM	Average RMSE
RED-CNN	29.7417	0.8814	13.2369
Wasserstein GAN	29.7508	0.8792	13.0817
Pix2Pix GAN	30.5894	0.8711	11.9008

Table 9.1: Comparison of our models based on PSNR, SSIM and RMSE

9.2 Qualitative Results

The qualitative analysis is done by sending the results to two doctors. We consider their inputs and understand that the images obtained are acceptable. We present details of the doctor review in the next chapter. In this section we look at the denoising results obtained by our models. The first set of results include a model-to-model visual comparison. In the top row the image on the left is the LDCT and the image on the right is NDCT. The lower row indicates the result obtained from the three models. Model-1 is the residual autoencoder, Model-2 is the W-GAN and Model-3 is the pix2pix GAN.

9.2.1 Visual Comparison of Model Ouputs

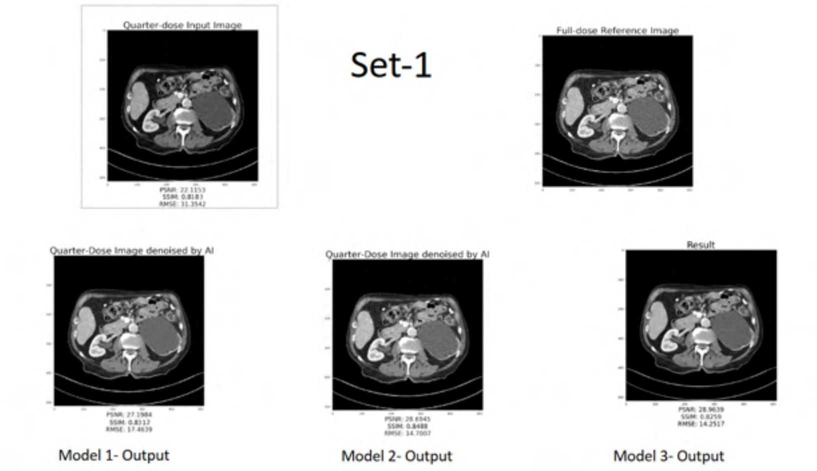


Figure 9.1: Results Set-1

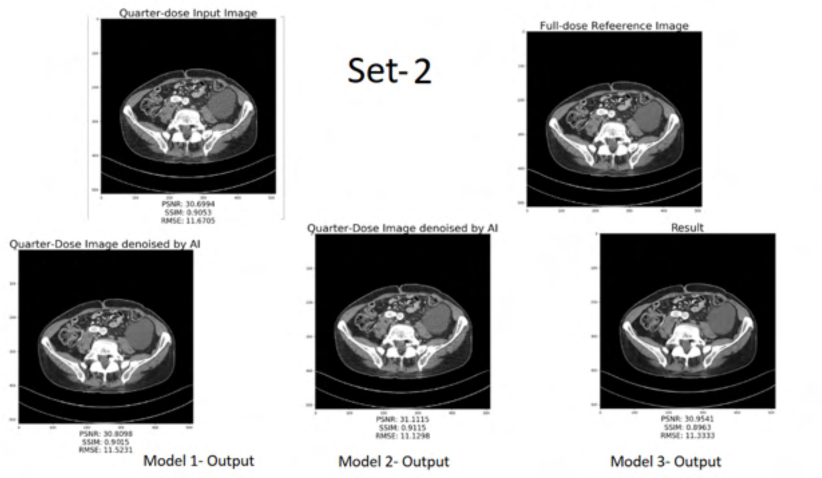


Figure 9.2: Results Set - 2

9.2.2 Individual Model Image Output

We consider the individual images in the coming section.

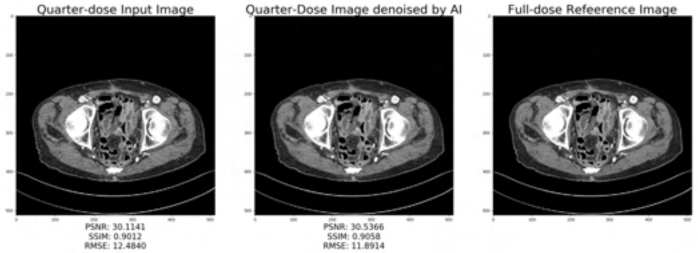


Figure 9.3: Red-CNN: Result 1

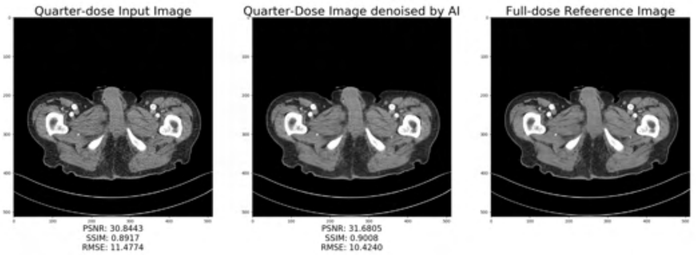


Figure 9.4: Red-CNN: Result 2

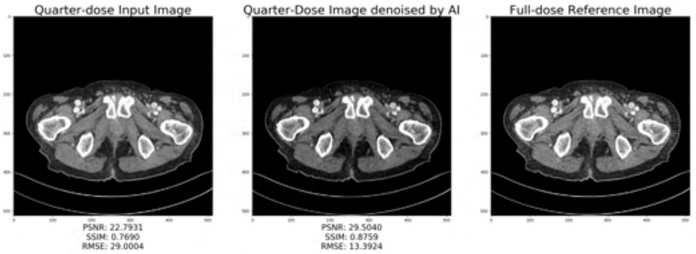


Figure 9.5: W-GAN: Result 1

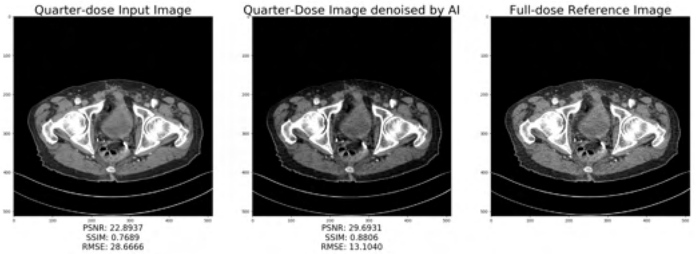


Figure 9.6: W-GAN: Result 2

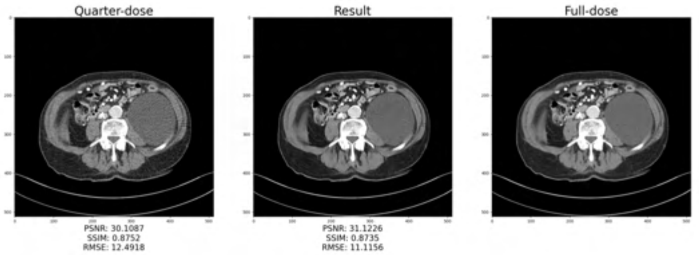


Figure 9.7: Pix2Pix: Result 1

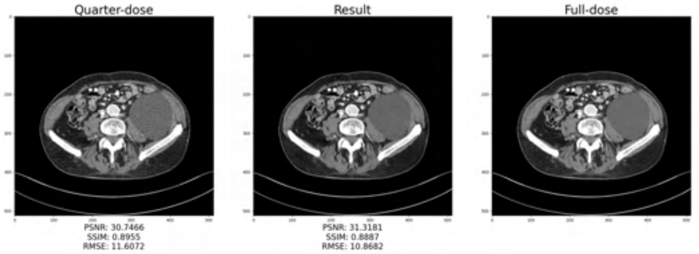


Figure 9.8: Pix2Pix: Result 2

Chapter 10

Doctor Review

We approached two distinguished radiologists, and tried to understand the problem from the medical point of view. We used their reviews of our project as part of the qualitative analysis.

The first review is given by Dr. Shailendra, Consultant Radiologist at Delta Dianostic Services. The second review is given by Dr. Sudheendraswamy, chief consultant radiologist at Bagalkot Scans and Diagnostics. We thank them for their kind gesture for agreeing to review the project and provide apt feedback to improve the project.

Doctor Review

[illegible]

[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	● [illegible] ● [illegible] ● [illegible]
[illegible]	[illegible]
[illegible]	● [illegible] ● [illegible]
[illegible]	[illegible]

[illegible]

[illegible]

[illegible]
Consultant Radiologist,
CT In-charge,
Delta Diagnostic Services,
Gandhi Bazaar Main Road,
Basavangudi,
Bangalore
Contact Details: Ph: +91 **99803 37274**
Email ID: **manshyl2004@gmail.com**

Doctor Review

I have reviewed the project titled "Denoising Low Dose CT images using generative models". I have the following remarks regarding the same.

Positives

1. The generated images work well and the features generated are acceptable. The quality of the generated images are around 75% comparable to the original image, and the features are visible.
2. The lowering of dosage also results in less power being used for the generation of the CT scan images, therefore, making it cost effective.
3. Lower dosage results is less ionising radiation to the patients and limits side effect of the CT scan.
4. The models mentioned in the above project have been analysed. I find model III to be the best performing amongst the three in terms of features retention. It shows more resemblance to full dose image in term of resolution, reduced artifacts and graininess (noise).

Negatives

1. CT images are dependent to some extent on the body mass index of the patient. The images considered belong to patients with normal BMIs, and the cases of overweight, underweight, children , elderly patients will need to be investigated further.
2. The number of patients considered while training is low. The project deals only with abdomen images. I would encourage that other body parts be used and supported.

Concluding remarks:

The project appears to have promising effect on diagnostic radiology. By using lower radiation dosage we are significantly reducing the harmful effects of the radiation. It also helps in repeatability of the study in needed cases. Reducing dosage , power utilisation can make this modality much cheap for the needy.

More models need to be established so that it can be adopted to routine diagnostic radiology practice in near future.

Dr. Sudheendraswamy V B.

Chief Consultant Radiologist;

Bagalkot scans and diagnostics,

Contact Details: Ph no: +919739862755

Email ID: drsudhee5@gmail.com

Chapter 11

Conclusion

We have done a comparative study with quantitative and qualitative analysis of generative models belonging to two classes of methods the autoencoder and Generative Adversarial Neural Networks to solve the problem of CT scan image denosing. We find that all the models are capable of denoising the LDCT image proving the viability of using deep learning for medical image denosing. We observed that the images produced by GANs have a higher PSNR values when compared to the autoencoder with pix2px performing the best as shown in table 9.1 implying that GANs are able to model the underlying noise model better and hence are better at suppressing it too while also maintaining the structural integrity of the image. This is further validated by the results of the qualitative review by Dr Sudheendraswamy who concluded that model-3 (pix2pix GAN) was the best performing model with its output resembling the NDCT image in terms of resolution, artifacts and graininess. The l1 reconstruction loss used to train the autoencoder does a good job of denoising the image but introduces blurring around edges in the image this is most likely because the l1 loss does averaging across all spatial locations and hence is unable to capture the high-frequency components of the image.

Furthermore this study serves as a proof of concept for the commercial viability of adopting generative models for denoising of low dose CT scan images and therefore help in limiting the harmful effect of the CT scan on patients due to the reduced effect of ionising radiation. The method can also help cut costs as the power required to run CT scan at lower dosage is significantly less.

Chapter 12

Future Work

Although the methods discussed show promising results there are a few problems that have to be addressed which we leave for future work as follows:

- Training on a larger low-dose CT dataset that was just publicly released.
- The current work concentrates on denoising of abdomen CT scans, denosing of other organs has to be explored under a unified framework.
- The variation in age and BMI of different patients has to be taken into account when producing the denoised result.

Bibliography

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

[2] Low Dose CT Grand Challenge. 2017. [Online]. Available:. *AAPM*, http://www.aapm.org/GrandChallenge/LowDoseCT/.

[3] 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.

[4] Christopher Olah. *Neural Networks, Manifolds, and Topology*.

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

[6] 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.

[7] 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.

[8] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

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

[10] 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.

[11] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[13] Hu Chen, Yi Zhang, Mannudeep K Kalra, Feng Lin, Yang Chen, Peixi Liao, Jiliu Zhou, and Ge Wang. Low-dose ct with a residual encoder-decoder convolutional neural network. *IEEE transactions on medical imaging*, 36(12):2524–2535, 2017.

[14] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

[15] Zhengwei Wang, Qi She, and Tomas E Ward. Generative adversarial networks: A survey and taxonomy. *arXiv preprint arXiv:1906.01529*, 2019.

[16] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.