UNIVERSITÉ DE BRETAGNE OCCIDENTALE

DOCTORAL THESIS

Tomographic Image Reconstruction with Neural Networks

Author:
Venkata Sai Sundar
KANDARPA

Supervisor:
Dr. Dimitris VISVIKIS,
Dr. Alexandre Bousse

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

LATIM
Biologie Santé

January 18, 2022

Declaration of Authorship

I, Venkata Sai Sundar KANDARPA, declare that this thesis titled, "Tomographic Image Reconstruction with Neural Networks" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"Thanks to my solid academic training, today I can write hundreds of words on virtually any topic without possessing a shred of information, which is how I got a good job in journalism."

Dave Barry

UNIVERSITÉ DE BRETAGNE OCCIDENTALE

Abstract

Biologie Santé
Biologie Santé

Doctor of Philosophy

Tomographic Image Reconstruction with Neural Networks

by Venkata Sai Sundar KANDARPA

Neural Networks are extensively used in the field of medical imaging for biomedical image segmentation, cancer diagnosis, image analysis, etc. The advancements in computation power (GPUs) and efficient memory utilization have propelled the spread of deep neural networks into various domains. The main motivation behind the use of neural network approaches is faster prediction (compared to traditional methods) without compromising on the quality of the result. Tomographic image reconstruction has also benefited from the development of neural networks. Medical image reconstruction involves the task of mapping raw measurement data collected by the detector to images that are comprehensible to a radiologist. A medical image reconstruction algorithm essentially approximates this mapping to predict the best possible image. There are established analytical and iterative reconstruction algorithms which have over the years proven to be effective in producing the best image possible. Convolutional neural networks (CNN) specifically have proven to be exceptional in tasks related to images such as denoising, deblurring, and super-resolution. The use of neural networks in Positron Emission Tomography (PET) and Computed Tomography (CT) reconstruction has been explored in this thesis. Novel frameworks called DUG-RECON (Double U-Net Generator) for PET, CT image reconstruction, and LRR-CED (Low-Resolution Reconstruction aware Convolutional Encoder-Decoder) for Sparse-view CT image reconstruction and Total-Body PET image reconstruction are proposed in this manuscript. Quantitative analysis of the images reconstructed with the proposed methods indicated that image quality was either better or on par with standard reconstruction algorithms.

Acknowledgements

The acknowledgments and the people to thank go here, don't forget to include your project advisor...

Contents

List of Figures

List of Tables

List of Abbreviations

LAH **L**ist **A**bbreviations **H**ere
WSF **W**hat (it) **S**tands **F**or

Physical Constants

Speed of Light $c_0 = 2.997\,924\,58 \times 10^8\,\mathrm{m\,s^{-1}}$ (exact)

List of Symbols

a	distance	m
P	power	W ($\mathrm{J\,s^{-1}}$)
ω	angular frequency	rad

For/Dedicated to/To my…

Introduction

0.1 Motivation

The use of deep learning in medical imaging has been on the rise over the last few years. It has widely been used in various tasks across medical imaging such as image segmentation (Ronneberger, Fischer, and Brox, 2015; Guo et al., 2019; Sinha and Dolz, 2019; Dolz et al., 2018; Hatt et al., 2018), image denoising (Kadimesetty et al., 2018; Li et al., 2020a; Chen et al., 2017; Yang et al., 2018), image analysis (Litjens et al., 2017; Amyar et al., 2019; Cui et al., 2018). Deep learning based algorithms produce faster results along with best possible quality in accordance with existing state of the art methods (Leuschner et al., 2021). Medical Image reconstruction too has benefited hugely with the advancement of deep learning (Reader et al., 2020; Zhang and Dong, 2020). Medical Image reconstruction corresponds to the task of mapping raw projection data retrieved from the detector to image domain data. During the course of this thesis, the focus has been towards PET and CT image reconstruction. Both these modalities present a unique of set of challenges for image reconstruction.

PET imaging is a form of emission tomography wherein the image reconstruction task revolves around identifying the radio-tracer distribution emitted from the patient. A PET image gives functional information about the organs in a patient making it invaluable for oncology. Some of the challenges in PET image reconstruction are scatter, attenuation and difficulty in identifying the exact annihilation point of the electron-positron. Despite being the most sensitive emission tomography modality, the number of photons captured is low relative to the photons emitted contributing to further image degradation. These challenges result in very noisy images when reconstructed with analytical algorithms. These challenges are addressed by Iterative/Model-based approaches which take into account detector geometry, noise statistics and approximate scatter and attenuation correction resulting in better image quality.

CT imaging on the other hand is an example of transmission tomography. The extent of attenuation undergone by X-Rays that pass through a patient are measured to obtain attenuation maps. In CT imaging research, there has been active interest in sparse-view and low-dose reconstruction scenarios. In both cases, severe artifacts are introduced in reconstructed images either due to incomplete projections or low counts. Many established model-based iterative methods account for the low-dose and sparse-view settings to remove artifacts and noise from the reconstruction (Nuyts et al., 1998; Elbakri and Fessler, 2002; Liu et al., 2013). However, these methods require the knowledge of the noise and artifacts statistics and generally have longer reconstruction times (Kim, Ramani, and Fessler, 2014).

The main tasks involved in image reconstruction can be broadly categorized into three: sinogram correction, domain translation from sinogram to image, and image correction. Algorithms either tackle each of the tasks individually or simultaneously account for them. One can relate to these tasks in the domain of computer vision wherein deep learning architectures have revolutionized the field by producing the state of the art results in most applications (Guo et al., 2016). For example, effective use of deep learning-based methods is seen in dealing with image denoising (Kadimesetty et al., 2018; Li et al., 2020a; Chen et al., 2017; Yang et al., 2018), super resolution (Ledig et al., 2017; Lim et al., 2017) and image-to-image translation tasks (Isola et al., 2017; Zhu et al., 2017). The continuous improvement in the availability of public data has further propelled interest in data-driven medical image reconstruction making it an active area of research. This thesis aims to explore novel deep learning approaches for PET and CT image reconstruction. Most common ways to introduce deep learning architectures in the image reconstruction pipeline are for pre-processing to correct raw projection data from the detector and post-processing to improve images reconstructed with existing methods. Another way is to embed the network into an iterative algorithm to enable faster convergence. The relatively less explored way called direct image reconstruction is to utilize neural networks alone for the entire reconstruction process. In this thesis two novel CNN-based methods are proposed and their performance is validated with both simulated data and real data.

0.2 Thesis Organization

This thesis is divided into seven chapters with the first two chapters giving a general introduction to image reconstruction and neural networks respectively. The third chapter presents the relevant literature review, wherein the application and impact of deep learning in image reconstruction research focused on PET and CT is presented. The next three chapters elaborate the different deep learning-based methods proposed in the thesis. In chapter 4, we discuss reconstruction framework DUG for PET and CT image reconstruction. A novel method for Sparse-view CT reconstruction called LRRCED is covered in chapter 5. A modified version of LRRCED for total body PET is discussed in chapter 6. Potential improvements and ideas for future work are presented in the final chapter.

Chapter 1

Image Reconstruction

Tomographic imaging is the process of observing an object through its cross-sections. It is a non-invasive technique where the interior of an object is visualized without any clinical intervention. In tomographic imaging usually a detector measures the radiation after it's interaction with the object. The measured data is transformed into comprehensible images that can be analyzed by the radiologist. This process of mapping measured data into images is called as image reconstruction. This chapter presents an introduction to the imaging principles of PET and CT. Analytic and model-based iterative reconstruction (MBIR) methods are then discussed both from a general standpoint and with algorithms specific to the respective imaging modality.

1.1 PET

PET images provide functional information to the radiologist making them invaluable in image analysis. The application of PET imaging has been on the rise in oncology, cardiology and neuropsychiatry. The increased application lead to the development of many novel reconstruction approaches targeting better image quality. PET is a form of emission tomography wherein the patient to be imaged emits radiation which is collected by a detector. This emission is a result of positron emitting radionuclide injected into the patient which causes positron-electron annihilation. Typical radio-tracers used in PET are ^{18}F-fludeoxyglucose (^{18}F-FDG), fluorothymidine (FLT), rubidium chloride, etc. Each of these radio-tracers is characterized by a positron emitting radio isotope. The positron decay for a radioactive nuclei (^{18}F for example) can be written as follows:

$$^{18}_{9}F \rightarrow {}^{o}_{+1}\beta + {}^{18}_{8}O$$

The positron emitted (${}^{o}_{+1}\beta$) is an unstable particle and it almost immediately annihilates with an electron. This annihilation results in the production of gamma photons that travel in opposite directions in accordance with the law of conservation of momentum. The simultaneous detection of these photons (also called coincidence events) enables the estimation of tracer distribution. The aim of image reconstruction in PET is to determine this tracer distribution. A PET scanner detects the coincidence events through a set of detectors arranged in a circular fashion. This design of the scanner facilitates detection of coincidence photons between a pair of detectors (d_p and d_q). The centers of two detectors are connected by a straight line called LOR. Photon pairs that are not subject to scatter are a result of annihilation events that occur along a thin volume surrounding the LOR. In PET, f is the distribution of a radiotracer delivered to the patient by injection, and is measured through the detection of pairs of γ-rays emitted in opposite directions (indirectly from the positron-emitting radiotracer).

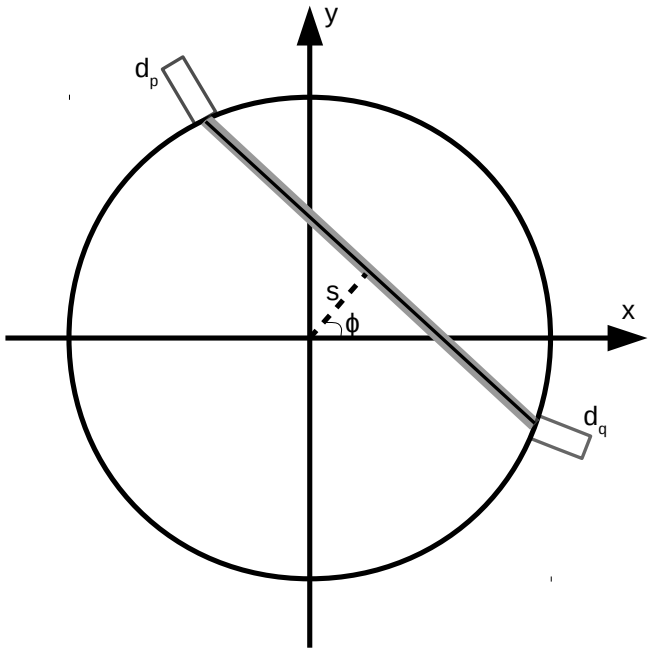


FIGURE 1.1: Depiction of a circular PET detector with detectors d_p and d_q connected with a LOR indicated in gray.

The number of detected coincidence events is related to the LOR (L_{d_p,d_q}) connecting the centers of detectors d_p and d_q through a sensitivity function $\psi(\vec{r} = (x, y, z))$. It is a Poisson variable whose mean can be written as:

$$\left\langle p_{d_p,d_q} \right\rangle = \tau \int_{\text{FOV}} d\vec{r} f(\vec{r}) \psi_{d_p,d_q}(\vec{r}) \tag{1.1}$$

where $f(\vec{r})$ denotes tracer concentration and τ is the acquisition time. The tracer concentration is assumed to be contained within the field of view (FOV). The reconstruction task can be summarized as estimating tracer concentration f, given measured data p_{d_p,d_q}, $(d_p, d_q) = 1 \dots N_{LOR}$. The above linear model is typically used by analytical algorithms. The measurement data is assumed to have been corrected for non-linear effects like scatter and random coincidences. Another approximation is that $\psi(\vec{r}) = 0$, except when $r \in L_{d_p,d_q}$. The measured data are therefore modeled as line integrals of tracer distribution (f):

$$\left\langle p_{d_p,d_q} \right\rangle = \int_{L_{d_p,d_q}} d\vec{r} f(\vec{r}) \tag{1.2}$$

The coincidences from the detector are typically rearranged either in list-mode or sinogram format. List mode data is a sequential recording of coincidence events. Time and energy of each detected photon can also be recorded. It has special significance in time of flight imaging for PET. Most analytical reconstruction algorithms on the other hand are tailor made for sinogram data format. Fig 1.1, represents a trans-axial slice of a PET scanner. One can model 2-D sinogram model with this representation. The variables s and ϕ are utilized to relate the LOR to the Cartesian co-ordinates (x, y). The radial variable s is the distance between the center of the detector ring and the LOR, while angular variable (ϕ) gives the orientation of the LOR. For a co-ordinate t along the line, Eq 1.2 now becomes:

$$p\,(s, \phi) = \int_{-\infty}^{\infty} dt f(x = s\cos\phi + t\sin\phi, y = s\sin\phi + t\cos\phi) \tag{1.3}$$

Through the line integral approximation and keeping in context the corrected PET data, $p_{d_a,d_b} \approx p(s, \phi)$. The function that maps the tracer distribution onto the line integrals is called as the x-ray transform. It is equivalent to the 2D version of the Radon transform.

1.2 CT

CT imaging is a form of transmission tomography. The high resolution images obtained from CT scans have many applications. They are extensively used in diagnosis of muscle, tissue and bone disorders. They serve a guide

for surgery planning and also to pin-point exact location of tumors. In emergency situations like a road accident, CT scan is utilized to check for internal bleeding. However, the radiation passed through the patient has been a topic of constant debate in this imaging modality. Research in recent times has been focusing on methodologies to reduce radiation while keeping the image quality intact.

A typical CT imaging setup consists of a X-ray source, the object to be imaged and detectors to measure the extent of attenuation experienced by the X-rays. When X-rays are passed through an object they suffer attenuation due to scatter and absorption. Scattering occurs when a X-ray photon dislodges an electron by transferring a part of it's energy. This phenomenon also called Compton scatter is depicted in Fig 1.2.

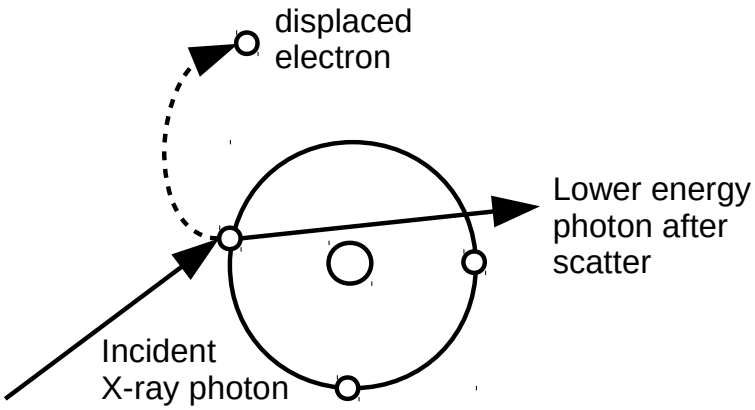


FIGURE 1.2: Depiction of Compton scatter

Complete absorption happens through photo-electric effect where the entire energy of the x-ray photon is transferred to the electron. The difference is seen in Fig 1.3, where the incident photon disappears after scatter.

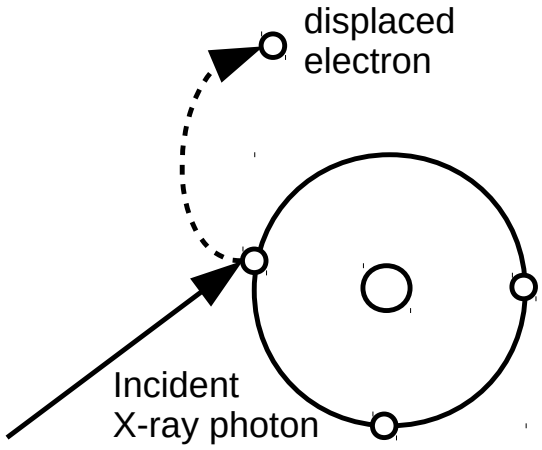


FIGURE 1.3: Depiction of Photo-electric effect

Different materials exhibit different absorption properties hence have unique linear attenuation co-efficient. Let the intensities of incident x-ray and the one after absorption be I_0 and I_a respectively. From Beer-Lambert's law, we have:

$$I_a = I_0 \cdot \exp(-p) \tag{1.4}$$

where p is the line integral of attenuation coefficients along the path of the x-ray photons. Similar to 1.2, measured data in CT can be modeled with line integrals p:

$$p = \ln \frac{I_a}{I_0} \tag{1.5}$$

The material specific property of attenuation (μ) varies with the energy of the incoming X-ray. It reduces with the increase in energy of the X-ray. A cross-sectional CT image consists of

Over the years many imaging geometries have been developed to maximize detector efficiency and obtain better image quality. The first generation of CT scanners consisted of X-ray beam source and a small detector that rotated and linearly translated around the patient. It had much longer scanning time compared to modern CT scanners. The second generation setup consisted of fan-beam source with an array of detectors. The motion was similar to that of the first generation. The third generation fan beam geometry is depicted in Fig 1.4, the motion was restricted to rotation of the source-detector setup. The fourth generation consisted of stationary circular array of detectors similar to PET with a rotating source.

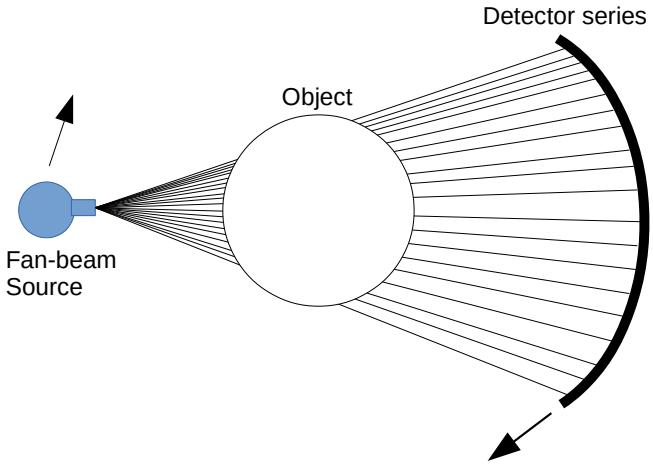


FIGURE 1.4: Fan-beam geometry: the source and the detector rotate around the object

A representation of a modern helical cone beam scanner is shown in Fig 1.5. The cone-beam source is rotated around the patient while the bed translates

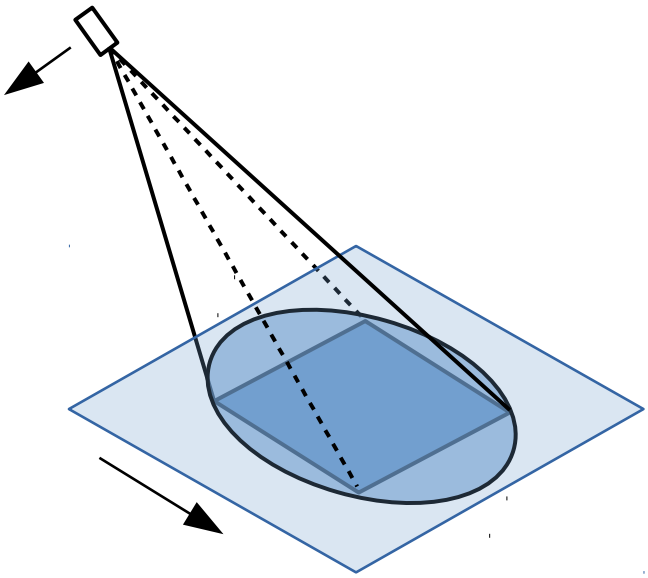


FIGURE 1.5: Cone-beam geometry: the source rotates around the patient while the bed is translated creating a helical scan.

linearly resulting in a helical orbit. The detector is a 2D array of crystals making it more efficient and faster for data acquisition.

1.3 Analytic Reconstruction

The starting point of analytic reconstruction is the central slice theorem. It states that 2D Fourier transform of the image (f) is related to the 1D Fourier transform of the x-ray transform as follows:

$$P(v,\phi) = f\left(v_x = v\cos\phi, v_y = v\sin\phi\right) \tag{1.6}$$

where

$$P(v,\phi) = (\mathcal{F}p)(v,\phi) = \int_{\mathbb{R}} dsp(s,\phi)\exp(-2\pi isv) \tag{1.7}$$

and v is the frequency variable associated with s. In the context of tomographic image reconstruction this theorem has the following implication: given the measurement data for all projection angles $\phi \in [0,\pi]$, the radial line sweeps all the frequencies hence making it possible to compute $f(v_x,v_y)$ for $(v_x,v_y) \in \mathbb{R}^2$. The image f can then be estimated by finding the inverse 2D Fourier transform.

FBP

One of the most used reconstruction algorithms across modalities is the filtered back-projection algorithm. The version with continuous sampling is written as follows:

$$f(x,y) = \left(X^* p^F\right)(x,y) = \int_0^\pi d\phi p^F(s = x\cos\phi + y\sin\phi, \phi) \tag{1.8}$$

where filtered projections p^F are given by

$$p^F(s,\phi) = \int_{-R_F}^{R_F} ds' p\left(s',\phi\right) h\left(s - s'\right) \tag{1.9}$$

and h is the ramp filter given by

$$h(s) = \int_{-\infty}^{\infty} dv|v| \exp(2\pi i s v) \tag{1.10}$$

The function mapping from p^F to λ is the back-projection operator. In reality discrete sampling is required to accurately model the acquisition process. The discrete implementation of the FBP can be written as follows:

$$\boldsymbol{x}(i,j) = \frac{\pi}{N_\phi} \sum_{l=0}^{N_\phi - 1} \boldsymbol{y}_f(s = i\cos\phi_l + j\sin\phi_l, \phi_l) \tag{1.11}$$

where x is the image for a set of pixels (i,j), $\boldsymbol{y}_f$ are the filtered projections obtained by filtering the projections, expressed in terms of radial variable s and projection angle ϕ, and N_ϕ number of projection angles. The above equation is the approximation of backprojection by a discrete quadrature.

1.4 Model-Based Image Reconstruction (MBIR)

Analytical methods are faster to implement and practical in a clinical setting but they are vulnerable to noise. The assumptions made in analytical formations are that the measurements are continuous and the solutions are of integral formulation. Sampling is done to the data a posteriori. They are also highly susceptible to system geometry. Since the 80's, MBIR techniques (Shepp and Vardi, 1982; Fessler, Sonka, and Fitzpatrick, 2000) became the standard approach. As they model the stochasticity of the system, they are more robust to noise as compared with FBP, and can be completed with a penalty term for additional control over the noise (De Pierro, 1995). They

also incorporate corrections for scatter and are independent of detector geometry.

1.4.1 Data Model for PET

The starting point of any model-based method is the data model. The measurement $\boldsymbol{y}$ is a random vector modeling the number of detection (photon counting) at each of the n detector bins, and follows a Poisson distribution with independent entries:

$$\boldsymbol{y} \sim \text{Poisson}(\bar{\boldsymbol{y}}(\boldsymbol{x})) \tag{1.12}$$

where $\bar{\boldsymbol{y}}(\boldsymbol{x}) \in \mathbb{R}^n$ is the expected number of counts (noiseless), which is a function of the image $\boldsymbol{x}$.

The expected number of counts is

$$\bar{\boldsymbol{y}}(\boldsymbol{\lambda}) = \boldsymbol{A}\boldsymbol{\lambda} \tag{1.13}$$

where $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ is a system matrix such that each entry $[\boldsymbol{A}]_{i,j}$ represents the probability that a photon pair emitted from voxel j. Image reconstruction is achieved by finding a suitable image $\hat{\boldsymbol{x}} = \hat{\boldsymbol{\lambda}}$ that approximately solves

$$\boldsymbol{y} = \bar{\boldsymbol{y}}(\boldsymbol{x})\,. \tag{1.14}$$

1.4.2 Data Model for CT

Let an image be represented by $\boldsymbol{x} \in \mathbb{R}^m$ and the scanner measurement by $\boldsymbol{b} \in \mathbb{R}^n$ where m is the number of voxels and n is the number of measurements. In 2-D CT imaging n depends on the number of detectors N_d and the number of angles N_a. The task of medical image reconstruction corresponds to finding a mapping from $\boldsymbol{b}$ to $\boldsymbol{x}$. The measurement $\boldsymbol{b}$ is a random vector modeling the number of detection (photon counting) at each of the n detector bins, and follows a Poisson distribution with independent entries, i.e.,

$$\boldsymbol{b} \sim \text{Poisson}(\bar{\boldsymbol{b}}(\boldsymbol{x})) \tag{1.15}$$

where, $\boldsymbol{b} = [b_1(\boldsymbol{x}), \ldots, b_n(\boldsymbol{x})]^\top \in \mathbb{R}^n$ and $\bar{\boldsymbol{b}}(\boldsymbol{x}) = [\bar{b}_1(\boldsymbol{x}), \ldots, \bar{b}_n(\boldsymbol{x})]^\top \in \mathbb{R}^n$ is the expected number of counts (noiseless), which is a function of the image $\boldsymbol{x}$.

The image $\boldsymbol{x} \in \mathbb{R}^m$ is a vectorized input image (also referred to as attenuation) representing the measure of X-rays absorbed or scattered as they pass through the patient. In a monochromatic setting, the expected number of counts $\bar{\boldsymbol{b}}(\boldsymbol{x})$ is given by the Beer-Lambert law, i.e.,

$$\bar{b}_i(\boldsymbol{x}) = I \cdot \exp(-[\boldsymbol{A}\boldsymbol{x}]_i) \quad \forall i = 1, \ldots, n \tag{1.16}$$

where, I is the intensity and $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ is a system matrix such that each entry $[\boldsymbol{A}]_{i,j}$ represents the contribution of the j-th image voxel to the i-th detector. Given the raw projections $\bar{\boldsymbol{b}}$, we take the logarithm as follows

$$y_i = \log\left(\frac{I}{\bar{b}_i}\right) \quad \forall i = 1, \ldots, n \tag{1.17}$$

where we assumed that the intensity I is sufficiently high so that $b_i > 0$ for all i. Image reconstruction is based on finding a suitable image $\hat{\boldsymbol{x}}$ that approximately solves

$$\boldsymbol{y} = \boldsymbol{A}\hat{\boldsymbol{x}} \tag{1.18}$$

where $\boldsymbol{y} = [y_1, \ldots, y_n]^\top \in \mathbb{R}^m$. The reconstruction can also be achieved with more sophisticated iterative techniques that account for the stochastic properties of the measurement (1.12) Nuyts et al., 1998; Elbakri and Fessler, 2002.

In a sparse-view setting, the number of rotation angles of the detector is decreased in order to reduce the radiation passing through the patient. This implies a reduction in the number of projection angles in the measurement $\boldsymbol{y}$.

1.4.3 Maximum Likelihood Expectation Maximization (MLEM)

The most famous model based method for PET image reconstruction is the maximum likelihood expectation-maximization (MLEM) algorithm. The cost function for MLEM is based on Poisson likelihood given as follows:

$$\Pr\{\bar{\boldsymbol{y}} \mid \boldsymbol{x}\} = \prod_{j=1}^{N_{LoR}} \exp\left(-\langle y_j \rangle\right) \langle y_j \rangle^{y_j} / y_j! \tag{1.19}$$

Putting 1.13 in 1.19, taking log and dropping terms that do not depend on unknown image x we get the cost function for MLEM,

$$Q(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) = \sum_{j=1}^{N_{LOR}} \left\{ -\sum_{i=1}^{m} \boldsymbol{A}_{j,i} x_i + y_j \log\left(\sum_{i=1}^{m} \boldsymbol{A}_{j,i} x_i\right) \right\} \tag{1.20}$$

where Q is the cost function and the definition of other variables is consistent from above. As long as the matrix $\boldsymbol{A}$ is singular, the above cost function remains convex and results in a unique image. The update step to map from the current estimate x^n to the next estimate x^{n+1} can be written as follows:

$$\boldsymbol{x}_i^{n+1} = x_i^n \frac{1}{\sum_{j'=1}^{N_{LoR}} \boldsymbol{A}_{j',i}} \sum_{j=1}^{N_{LOR}} \boldsymbol{A}_{j,i} \frac{y_j}{\sum_{i'=1}^{m} \boldsymbol{A}_{j,i'} x_{i'}^n} \quad i = 1, \ldots, m \tag{1.21}$$

The initial estimate x_i^1, $i = 1, \ldots, m$ typically follows a uniform distribution. The denominator with sum over index i' is the forward projection operation. Hence it estimates the measured data for the current image estimate. The numerator with sum over index j is the back projection over the ratio of measured and estimated data. The MLEM algorithm does not include a prior and it converges to the image that best fits the data. This estimate has inherent instabilities as the fitting is done closely to the noisy measured data. Contrary to FBP, data corrections in MLEM can be added to the cost function. Attenuation correction can be introduced as a multiplicative factor while scatter and randoms are added to the cost function as follows:

$$\begin{aligned} &\boldsymbol{x}_i^{n+1} = x_i^n \frac{1}{\sum_{j'=1}^{N_{LOR}} \boldsymbol{A}_{j',i}/\alpha_{j'}} \sum_{j=1}^{N_{LoR}} \boldsymbol{A}_{j,i} \frac{y_j + 2\bar{r}_j + \bar{s}_j}{\sum_{i'=1}^{m} \boldsymbol{A}_{j,i'} x_{i'}^n + \alpha_j \left(2\bar{r}_j + \bar{s}_j\right)} \\ &i = 1, \ldots, m \end{aligned} \tag{1.22}$$

where y_j are measured data corrected for random and scatter, α_j is the factor for attenuation correction, $\bar{r}_j$ and $\bar{s}_j$ are estimates for random and scatter background in LOR j.

1.4.4 Ordered Subsets Expectation Maximization (OSEM)

The ordered-subsets expectation-maximisation (OSEM) algorithm is a modification of the MLEM algorithm which made it computationally practical for implementation in clinical setting. The LOR data is divided into $\boldsymbol{S}$ disjoint subsets: $J_1, \cdots, J_S \subset [1, \cdots, N_{\text{LOR}}]$. Each of the parallel projection is assigned to a unique subset: $c, c+S, c+2S, \cdots, \leq N_\phi$ to the subset J_{c+1}. MLEM (Eqn 1.21) is applied to each of the subsets individually in an orderly fashion. Subset $J_{N \bmod S}$ is used at iteration N:

$$\begin{aligned} &\boldsymbol{x}_i^{N+1} = \boldsymbol{x}_i^N \frac{1}{\sum_{j' \in J_{N \bmod S}} \boldsymbol{A}_{j',i}} \sum_{j \in J_{N \bmod S}} \boldsymbol{A}_{j,i} \frac{y_j}{\sum_{i'=1}^{m} \boldsymbol{A}_{j,i'} \boldsymbol{x}_{i'}^N} \\ &i = 1, \ldots, m \end{aligned} \tag{1.23}$$

1.4.5 Weighted Least Squares (WLS)

One of the most common iterative techniques for CT image reconstruction is the weighted least squared (WLS) method, the image $\hat{x}$ is estimated by minimizing the following:

$$\hat{x} = \arg\min_{x \succeq 0} \frac{1}{2} \|y - Ax\|_W^2 \tag{1.24}$$

where $W = \text{diag}\{w_i\}$ is the diagonal weighting matrix that constitutes for the variance of each ray and $\|z\|_W^2 = z'Wz$. Despite the statistical weighting, due to the noise in measurements and ill-conditioned problem of image reconstruction the image estimate will still be noisy.

1.4.6 Penalized MBIR

An improvement to the above mentioned MBIR algorithms can be brought by finding a balance between the desired a priori characteristics of the image and the data fitting. This balance is realized through a regularized cost function.

Absolute Difference

One such form of regularization is introduced through an edge preserving regularizer that penalized the differences between neighboring voxels:

$$R(x) = \sum_{j=1}^{n_p} \sum_{k \in \mathcal{N}_j} \psi_{jk} \left(x_j - x_k \right) \tag{1.25}$$

where $R(x)$ is the regularization term, that enforces piece-wise smoothness on the image and β the regularization parameter. ψ is a potential function that controls the penalization of differences in the neighboring voxels. Adding the penalty term in the cost function of WLS we get:

$$\hat{x} = \arg\min_{x \succeq 0} \Psi(x), \quad \Psi(x) \triangleq \frac{1}{2} \|y - Ax\|_W^2 + \beta R(x) \tag{1.26}$$

where N_j are the set of neighboring indices of the j^{th} voxel.

Total Variation (TV)

Complex image reconstruction problems like sparse-view CT are under-determined due to the limited number of projection data available for reconstruction. In

such a scenario stronger forms of regularization like total variation (TV) are utilized. Consider a 2D digital image $x[p, q]$, discrete form of TV can be written as:

$$\sum_{p}\sum_{p} |x[p, q] - x[p-1, q]| + |x[p, q] - x[p, q-1]| \tag{1.27}$$

This is an an-isotropic version of TV regularization. Images reconstructed with TV, sometimes have inexplicable artifacts due to the fact that the absolute value potential is not differentiable at 0.

Chapter 2

Neural Networks

Neural networks, also known as artificial neural networks (ANN) are machine learning algorithms that form the basis of deep learning. They inherit name and structure from neurons in the brain. Biological neurons transmit signals to one another through complex networks. This interconnected networking is realized though various combinations of neurons forming an ANN. Sets of artificial neurons are stacked on top of each other to form a layer. A typical neural network consists of many such layers that are connected to each other. The first layer is called input layer, the final layer is termed output layer and the layers in-between are called hidden layers. A neural network with three hidden layers is depicted in Fig 2.1. The transmission of data across the nodes or artificial neurons happens through the connections. Each and every node has a specific weight and threshold associated. The output from a node is passed through the connection only if the value is above the threshold. Neural network approaches are data-driven. Their performance improves as they learn through training on a dataset.

To further understand the working of a neural network, we can imagine each node to be solving the problem of linear regression. For example consider a node with four inputs ($x_i, i = 1, 2, \ldots 4$), four weights ($w_i, i = 1, 2, \ldots 4$) and a bias:

$$\sum_{i=1}^{m} w_i x_i + \text{ bias } = w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + \text{ bias} \tag{2.1}$$

The output of the node is the above summation after going through an activation function g:

$$\text{output } = g(x) = \begin{cases} 1 \text{ if } \sum w_1 x_1 + b \geq 0 \\ 0 \text{ if } \sum w_1 x_1 + b < 0 \end{cases} \tag{2.2}$$

In the above example, the given activation function of this node propagates the value 1 only when the weighted sum of it's inputs is non-negative. When

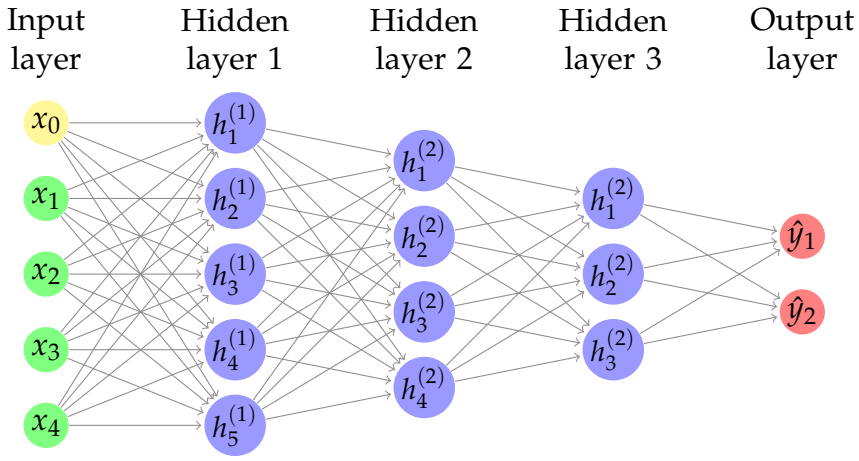


FIGURE 2.1: Depiction of a neural network with an input layer, three hidden layers and an output layer

the condition of an activation function are met, the output of this node becomes an input to the node to which it's connected. Due to the process of forwarding values through a network, an ANN is also called feed-forward network. Complex networks with multiple layers of these nodes are used in practical tasks. An important category of machine learning task is supervised learning. It involves training a neural network on labeled datasets. The goal of training a neural network is to minimize a cost function that enforces the closeness of predicted and real output labels. During the training the network reorganizes it's weights based on the loss function. This process of updating weights is called optimization. Each update is aimed at reaching a minimum of the loss function. A popular optimization method is gradient descent. It guides the model in the direction of reducing errors to reach an optima. The development of back-propagation (Rumelhart, Hinton, and Williams, 1986) has been instrumental in successful implementation of optimization algorithms for neural networks. A machine learning algorithm is typically specified by a cost function, an optimization procedure and a model. Even neural network design is based on these principles. One can find a co-relation between the iterative reconstruction algorithms based on gradient-based optimization and neural network training with gradient descent. It is to be noted that the non-linearity in the activation functions causes moss functions to become non-convex. This implies that gradient-based optimizers used for neural network training essentially drive the cost function to a very small value without a global convergence guarantee. Neural networks are initialized to small random variables prior to training as gradient descent without the convergence guarantee is sensitive to values of initial weights.

2.1 Cost Function

Neural networks can be represented by parametric models that define a distribution $p(\boldsymbol{y} \mid \boldsymbol{x}; \boldsymbol{\theta})$. The aim is to learn a conditional distribution to predict $\boldsymbol{y}$ given $\boldsymbol{x}$. Through principle of maximum likelihood, cross-entropy between the training data and the model's predictions become the cost function. This negative log-likelihood or cross-entropy between training data and model distribution can be written as:

$$\mathbf{J}(\boldsymbol{\theta}) = -\mathbb{E}_{\mathbf{x},\mathbf{y}\sim\hat{p}_{\text{data}}} \log p_{\text{model}}(\boldsymbol{y} \mid \boldsymbol{x}) \tag{2.3}$$

Given a specific p_{model}, the cost function exhibits a different form. Expanding the above generates some terms which are discarded as they do not depend on trainable model parameters. As an example, if p_{model} follows a Gaussian distribution $\mathcal{N}(\boldsymbol{y}; f(\boldsymbol{x}; \boldsymbol{\theta}), \boldsymbol{I})$, Equation 2.3 becomes:

$$\mathbf{J}(\theta) = \frac{1}{2}\mathbb{E}_{\mathbf{x},\mathbf{y}\sim\hat{p}_{\text{data}}} \|\boldsymbol{y} - f(\boldsymbol{x}; \boldsymbol{\theta})\|^2 \tag{2.4}$$

The above is equivalent to mean squared error (MSE) between the model distribution and the training data and is one of the most commonly used loss functions in training neural networks for linear regression. This approach of deriving the cost function from maximum likelihood removes the difficulty of choosing cost functions for each model. Choice of the model itself determines the cost function. Another popular loss function mean absolute error (MAE) can be derived from 2.3 by assuming p_{model} to follow a Laplacian distribution.

2.2 Output Unit

Neural networks as described above consists of an output layer after a series of hidden layers. The choice of cost function and output layer are highly dependent on each other. The representation of the output, determines the cross-entropy function. Given a set of hidden features defined by $\boldsymbol{h} = f(x; \boldsymbol{\theta})$., the role of the output layer is to transform the features appropriate for the task at hand. The most common choices for output layers are linear units and sigmoid units. Given a set of features h, a linear layer outputs a vector $\hat{\boldsymbol{y}} = \boldsymbol{W}^\top \boldsymbol{h} + \boldsymbol{b}$. A modification of a linear layer is rectified linear unit (ReLU) given by $g(z) = \max\{0, z\}$. The frequent usage of linear units is to find the mean of a conditioned Gaussian distribution. For regression tasks the output

unit typically has the linear activation. Tasks like binary classification require to define Bernoulli distribution for the maximum likelihood approach. The network needs to predict only $\boldsymbol{P}(y = 1 \mid x)$. The output value need to be in the interval $[0, 1]$. In this scenario a sigmoid activation does the task of transforming the hidden features into normalized probability value in the range $[0, 1]$. A sigmoid output unit is defined by:

$$\hat{y} = \sigma\left(\boldsymbol{w}^\top \boldsymbol{h} + b\right) \tag{2.5}$$

where σ is given by:

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \tag{2.6}$$

The hidden units are usually preferred to have ReLU or variations of ReLU as the activation in order to have significant gradients during optimization.

2.3 Backpropagation

Consider a feedforward network with an input $\boldsymbol{x}$ that produces an output $\boldsymbol{y}$. The propagation through the network starts with initial information from the inputs and continues through the hidden units at each layer, finally resulting in the output $\hat{\boldsymbol{y}}$. This process is termed as forward propagation. Back-propagation on the other hand makes computes the gradient by making the cost flow backwards through the network. Forward propagation is carried on during training to produce a scalar cost $J(\boldsymbol{\theta})$, which is then utilized by back-propagation to compute the gradients. Back-propagation is a simplified way for computing the gradients and is used with an optimization algorithm like stochastic gradient descent for network training. The most important gradient required in learning algorithms is the one of cost function with respect to learning parameters, $\nabla J(\boldsymbol{\theta})$.

The neural network given in Fig 2.1 follows computational graph representation. In order to discuss back-propagation, we formulate a simple notation using graphs. Each node in the graph can be considered to be a variable. The variable could be of any type, say a scalar, vector or a matrix. Another component of a computational graph is an operation. It is just a simple function based on one or more variables. An operation is assumed to return a

single output variable, which could have single or multiple entries. A simple computational graph with one hidden layer and sigmoid output unit is shown in Fig 2.2.

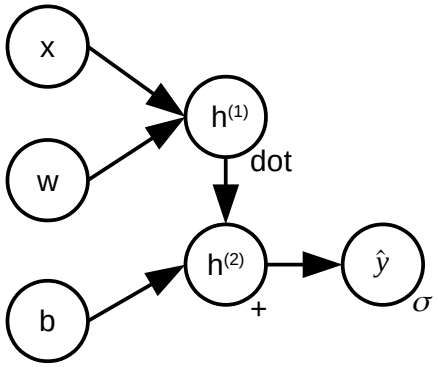


FIGURE 2.2: Computational graph with one hidden layer. The nodes in the first layer store the input x, weight w and the bias b. The second layer contains the hidden layer with 2 units each with the corresponding operation written below. The final layer is the output layer denoted by $\hat{y} = \sigma(wx + b)$, where σ is the sigmoid function defined earlier.

The gradients in the backpropagation algorithm are calculated by recursively applying the chain rule of calculus. The chain rule is a process of computing derivatives of functions based on multiple functions whose derivatives are already known. Back-propagation is an efficient implementation of chain rule with an order of operations feasible for computation. Let the input x,b and w to be real numbers, and h^1,h^2 and $\hat{y}$ be functions mapping from one real number to another, the chain rule can be written as follows:

$$\frac{dy}{dx} = \frac{dy}{dh^2}\frac{dh^2}{dh^1}\frac{dh^1}{dx} \tag{2.7}$$

where $h^1 = xw$, $h^2 = h^1 + b$ from Figure 2.2. We can generalize the above for a vector case with $x, w, b \in \mathbb{R}^m$ as follows:

$$\frac{\partial y}{\partial x_i} = \sum_j \frac{\partial y}{\partial h_j^1}\frac{\partial h_j^2}{\partial h_j^1}\frac{\partial h_j^1}{\partial x_j} \tag{2.8}$$

The chain rule involves many repeatable expressions which may need to be stored to avoid multiple re-computations for estimating gradients. Especially for complex neural networks it would lead to an exponentially high number of computations. A simplistic version of the backpropagation algorithm for a fully-connected multi-layer perceptron (MLP) is discussed in this section. For a supervised loss function $L(\hat{\boldsymbol{y}}, \boldsymbol{y})$, where $\hat{\boldsymbol{y}}$ is the predicted

output and $\boldsymbol{y}$ the target, forward propagation for a single training example is shown in Algorithm 1. After the forward propagation the gradient on the cost function J is calculated and then propagated through the network through back-propagation described in Algorithm 2.

Algorithm 1: Forward propagation algorithm for a single input example $\boldsymbol{x}$

Number of layers, l ;
Network weights represented by matrices, $\boldsymbol{W}^{(i)}, i \in \{1, \ldots, l\}$;
Bias parameters, $\boldsymbol{b}^{(i)}, i \in \{1, \ldots, l\}$;
Hidden units, $h^{(i)}, i \in \{1, \ldots, n\}$;
$h^{(0)} = \boldsymbol{x}$, ▷ Initializing input nodes ;
for $j = 1, \ldots, l$ **do**
  $a^{(j)} = b^{(j)} + \boldsymbol{W}^{(j)} h^{(j-1)}$ ▷ information from previous layers;
  $h^{(j)} = f(a^{(j)})$ ▷ activation in the current layer;
end
$\hat{\boldsymbol{y}} = h^{(l)}$;
$J = L(\hat{\boldsymbol{y}}, \boldsymbol{y}) + \lambda R(\theta)$ ▷ Cost function with a regularization ;

Algorithm 2: Backward propagation for neural network from Algorithm 1

Computing gradient g of the output layer;
$g = \nabla_{\hat{y}} J = \nabla_{\hat{y}} L(\hat{\boldsymbol{y}}, y)$
for $j = l, l-1 \ldots, 1$ **do**
  The gradient of a particular layer's output needs to be converted into gradient before activation f;
  $g = \nabla_{\boldsymbol{a}^{(j)}} J = g \circ f^{'}(a^{(j)})$;
  Gradients on weights and biases ;
  $\nabla_{b^{(j)}} J = g + \lambda \nabla_{b^{(j)}} R(\theta)$;
  $\nabla_{\boldsymbol{W}^{(j)}} J = g h^{(j-1)} + \lambda \nabla_{\boldsymbol{W}^{(j)}} R(\theta)$;
  Propagating the gradients through the preceding lower lowel activations;
  $g = \nabla_h J = \boldsymbol{W}^{(j)\intercal} g$
end

2.4 Optimization

Once the gradients are calculated through backpropagation algorithm, optimization procedures like gradient descent can be use to update the network

parameters (θ). The two algorithms in the previous section were demonstrated for a single example. In reality neural networks are often trained in parallel on multiple examples. This set of combined examples is called a batch and optimization algorithms are implemented accordingly for training in batches. In this section we discuss two of the most used optimization algorithms stocastic gradient dscent (SGD) and ADAM.

2.4.1 Stochastic Gradient Descent

SGD is an implementation of the popular gradient descent algorithm for training in batches. We obtain an estimate of the gradient by averaging the gradient over a minibatch of m training examples taken from the data distribution. SGD is depicted in Algorithm 3.

Algorithm 3: Training update at an iteration j for stocastic gradient dscent (SGD)

Learning rate ϵ_j;
Current parameters θ_k;
while *stopping criterion is not reached* **do**
  From the training set, sample m minibatch of examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ and corresponding targets $\{\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(m)}\}$
  Computing average gradient:
  $\hat{g} = \frac{1}{m} \nabla_{\theta_j} \sum_{i=1}^{m} L(f(\boldsymbol{x}^{(i)}, \theta_j), \boldsymbol{y}^{(i)})$
  Update: $\theta_j = \theta_j - \epsilon \hat{g}$
end

The learning rate ϵ_j is gradually decreased as the training progresses, due to the noise introduced by random sampling of minibatches.

2.4.2 Adam

Adam is another optimization algorithm which incorporates adaptive learning rate and momentum for faster convergence (Kingma and Ba, 2014). Momentum introduces velocity denoted by v that indicates speed and direction for parameters to update through parameter space. It is typically set to an exponentially decaying average of the negative gradient. Adam is derived from adaptive momentum. It is depicted in Algorithm 4.

Algorithm 4: Adam algorithm

Step size ϵ default usually 0.001 ;
Exponential decay rates ρ_1 and ρ_2, typically set to 0.9 and 0.999 ;
Constant δ, a very small number for stabilization, usually in the order of 10^{-8} ;
Parameters θ ;
1st and 2nd moment variables, initialized to $s = 0, r = 0$;
Time step $t = 0$;
while *stopping criterion is not reached* **do**
  From the training set, sample m minibatch of examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ and corresponding targets $\{\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(m)}\}$
  Computing average gradient:
  $\hat{g} = \frac{1}{m} \nabla_{\theta_j} \sum_{i=1}^{m} L(f(\boldsymbol{x}^{(i)}, \theta_j), \boldsymbol{y}^{(i)})$
  $t = t + 1$
  Update first momentum estimate: $s = \rho_1 s + (1 - \rho_1) g$
  Update second momentum estimate: $r = \rho_2 r + (1 - \rho_2) g \circ g$
  Correct bias in first moment: $\hat{s} = \frac{s}{1 - \rho_1^t}$
  Correct bias in second moment: $\hat{r} = \frac{r}{1 - \rho_2^t}$
  Calculate parameter update: $\Delta\theta_j = -\epsilon \frac{\hat{s}}{\sqrt{\hat{r} + \delta}}$
  Update: $\theta = \theta + \Delta\theta$
end

2.4.3 Universal Approximation Theorem

The wide usage of neural networks is a testimony of their ability to adapt across multiple applications. This is based on the universal approximation theorem which states that a feed-forward network with a linear output layer and at least one hidden layer with a non-linear squashing activation function (like sigmoid) can approximate any function mapping from any finite dimensional discrete space to another provided that the network has enough hidden units (Hornik, Stinchcombe, and White, 1990). This statement needs to be taken with a pinch of salt as it does not guarantee determining the optimal parameters of the network. It merely acknowledges the existence of a network that can represent the function in question. Training the network has two major challenges. One, the optimization process involved in training the network may not be able to find the network weights suitable to represent the function due to inadequate data (under-fitting problem). Two, the training could lead to a set of parameters that do not generalize well for the test data (over-fitting). Depending on the application and the data, network design is subject to change. The best network parameters that generalize well are usually obtained empirically through careful and logical experimentation.

In theory a network with a single layer is sufficient to learn the representation but it would need to be very large and therefore may fail to generalize. Hence, deeper architectures with multiple hidden layers are preferred over shallow network with infeasible number of neurons. In the next section, specialized deep neural network suitable for images called convolutional neural network (CNN) is discussed.

2.5 Convolutional Neural Network

The neural network depicted in 2.1 is an example of densely connected network, where all the neighboring nodes are connected with one another. As the size of data increases (say large image data), and the network becomes more complex, the number of parameters increases exponentially. To address this and also to be more suitable for image data CNNs were formulated. CNNs are extensively used in computer vision tasks like image classification, object detection, image segmentation (Voulodimos et al., 2018). The three main building blocks of a CNN are Convolution, Activation and Pooling. Each of these layers are discussed below:

2.5.1 Convolution

Images are digitally stored in the form of 2D or 3D matrices depending on the format. A convolution kernel (also known as filter) is a matrix that operates on these images and transforms them based on the kernel values. These kernel values are also known as weights in the neural network terminology. Typically, the size of the kernel is much smaller than that of the image. Many sets of these kernels form the convolution layer of the CNN. The movement of the kernel over the image can be made either by a single pixel or multiple pixels. This step size is called stride (s). The resulting output of a convolution between filter and image is called a feature map. Consider a kernel h and input image f with m rows and n columns. Convolution between h and f results in a feature map g:

$$g[m,n] = (h * f)[m,n] = \sum_i \sum_j h[i,j] f[m-i, n-j] \tag{2.9}$$

Given in Fig 2.3 is a representation of the convolution operation. Zero padding is used to manipulate the dimensions of the feature maps. In the above Figure above it is indicated with dotted lines. The function of padding

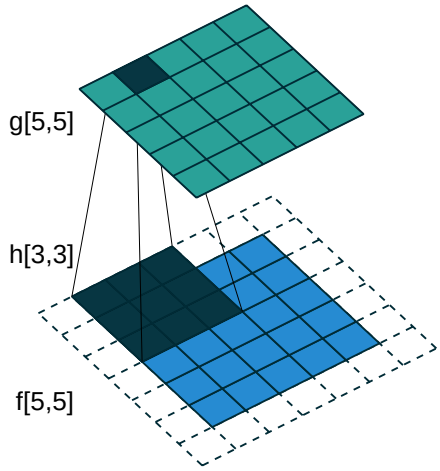


FIGURE 2.3: Convolution of an input image of dimensions 5×5 with a filter of dimensions 3×3.(Dumoulin and Visin, 2016)

here is to maintain same dimensions in the input image f and the feature map g. A CNN learns features from the input through many convolutional layers. The earlier layers learn general features like edges, contrast, the deep layers learn more abstract and finer details.

2.5.2 Activation Layer

The activation layer that follows the convolution layer in a CNN is most commonly the ReLU activation function, depicted in Fig 2.4.

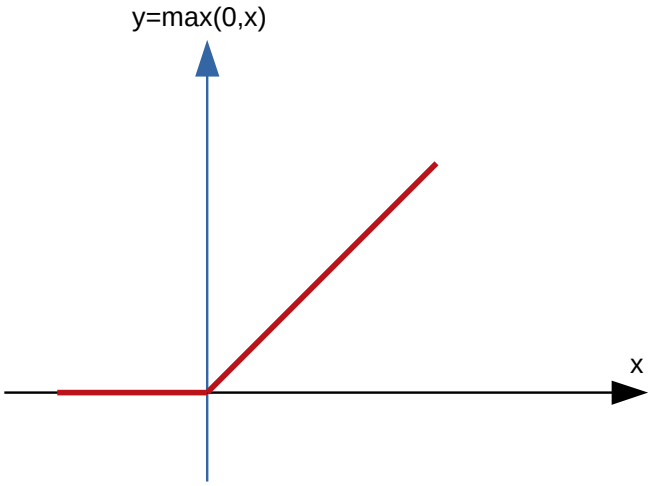


FIGURE 2.4: The ReLU function

Many of the task based on images are non-linear in nature. Whether a computer vision task like identifying objects in an image or a medical imaging task involving tumor detection, the relationships are far from being linear. The ReLU function increases this required non-linearity in the CNN.

2.5.3 Pooling Layer

The third building block of a CNN is the pooling layer. Pooling operation is mainly used to reduce the dimensions of a tensor which enables faster computation. Max pooling is the most commonly used pooling operation. A max pooling operator of a particular size returns the maximum value of a selected region in the feature map. Similar to a filter it is implemented with a specific stride. A max pooling filter with $s = 2$ is depicted in Fig 2.5.

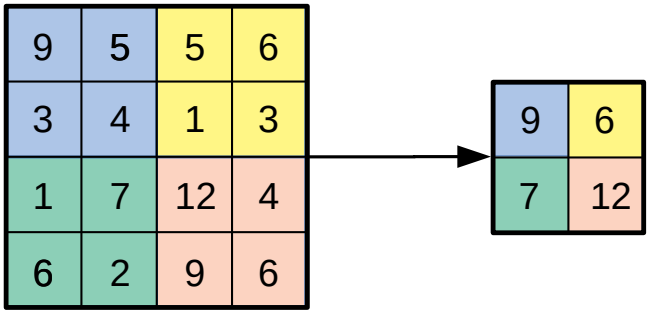


FIGURE 2.5: Max pooling with 2×2 filter and stride 1

A CNN with 2 convolutional layers, 2 activation layers and 2 pooling layers is represented in Fig 2.6. Layer number is given by l. The first and the last layer are the input and output respectively. Usually the last set of layers in a CNN used for classification, regression tasks is a fully-connected layer which is similar to the neural network represented in Fig 2.1. With the advent of powerful computation tools and efficient parallel processing, neural networks with many layers could be implemented. The term deep learning was coined for networks with this "deep" design (LeCun, Bengio, and Hinton, 2015). Deep neural networks could be trained over large datasets and they outperformed many existing state of the art algorithms in computer vision. In this thesis we focus specifically on CNNs under the umbrella of deep neural networks.

2.5.4 Neural Networks for Image to Image Translation

Image to image translation tasks require the CNN to map from image in one domain to an image in another related domain. This requires the design of the CNN to be quite different from the one depicted in Fig 2.6. Convolution and pooling operations compress the input to obtain an abstract representation in lower dimensions. This lower dimensional encoding is transformed back into an image through the use of transposed convolution operators. In contrast to the compressing nature of convolutions, they expand

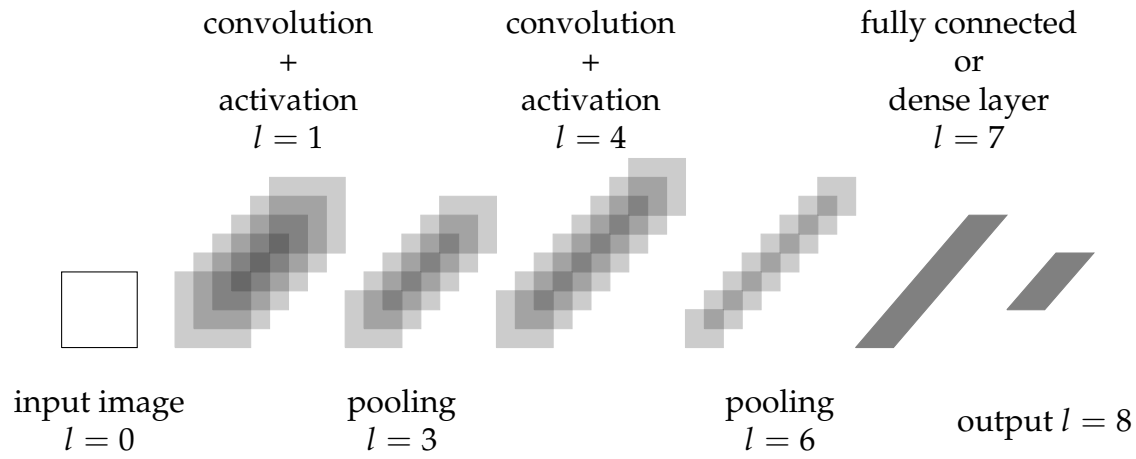


FIGURE 2.6: Architecture of a typical CNN. This representation was first proposed by LeCun and Bengio, 1995.

the input feature map. The combination of convolution+pooling and transposed convolutions are adjusted depending on the dimensions of the input and output images. Transposed convolution is shown in Fig 2.7. Essentially the transposed convolution spatially reverses the dimensions of the convolution+pooling operation. These networks with two parts where one part downsamples the input image and the other part upsamples the encoding into an image are called encoder-decoder networks. Since we use convolutions for achieving this encoder-decoder setup they are specifically called as convolutional encoder decoder (CED).

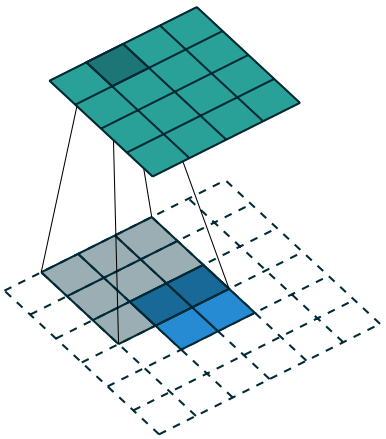


FIGURE 2.7: Transposed convolution over a 2×2 input to get a 4×4 output. (Dumoulin and Visin, 2016)

CEDs are used in a variety of image to image translation tasks. Across the literature one would find many variations used in super resolution, image segmentation, denoising and image reconstruction. This subset of CNNs

appropriate for image reconstruction task is represented in Fig 2.8.

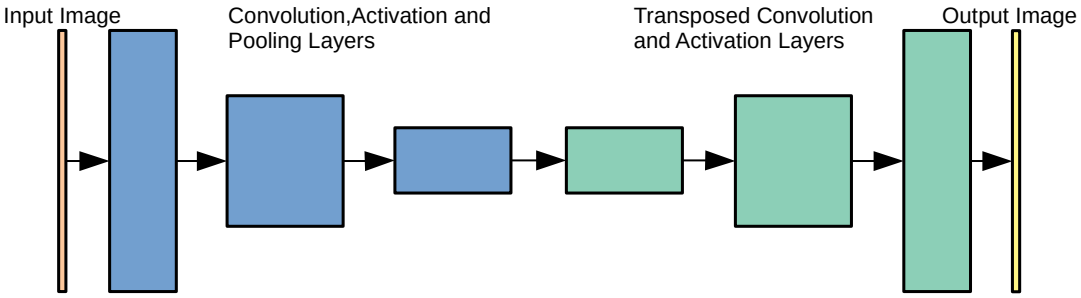


FIGURE 2.8: CNN for image to image translation tasks. This example has an identical structure in convolution path and the transposed convolution path.

The building blocks described in this section essentially form the basis of neural network approaches proposed in this thesis. The next chapter consists of a review of existing works in deep learning applied to medical image reconstruction. And the following chapters describe the proposed methods.

Chapter 3

Deep Learning and Tomographic Image Reconstruction

The impact of deep learning has been immense over the last few years in the field of medical imaging (Litjens et al., 2017; Greenspan, Van Ginneken, and Summers, 2016). Medical image reconstruction has also benefited hugely from the various advances in neural network architectures Wang, Ye, and De Man, 2020; Yedder, Cardoen, and Hamarneh, 2021; Reader et al., 2020. In the specific case of CT image reconstruction, there has been active interest in sparse-view and low-dose reconstruction scenarios. While with PET reconstruction on the other hand, low-dose imaging and total body imaging have been on the forefront. In both cases, obtaining high quality reconstructed images is a challenging task. Many established model-based iterative methods account for the low-dose and sparse-view settings to remove artifacts and noise from the reconstruction (Nuyts et al., 1998; Elbakri and Fessler, 2002; Liu et al., 2013). However, these methods require the knowledge of noise and artifacts statistics and generally have longer reconstruction times Kim, Ramani, and Fessler, 2014. Deep learning-based methods on the other hand are claimed to achieve reconstructed images with quality on par with iterative techniques and in a much shorter time frame Leuschner et al., 2021. In this work, the focus has been on CT and PET image reconstruction. Image reconstruction corresponds to the task of mapping raw projection data retrieved from the detector to image domain data. As depicted in Fig 3.1, one can broadly identify three different categories of approaches for the implementation of deep learning within the framework of medical image reconstruction:

(i) methods that use deep learning as an image processing step that improves the quality of the raw data and/or the reconstructed image (Gong et al., 2018; Maier et al., 2018);

(ii) methods that embed deep-learning image processing techniques in the iterative reconstruction framework to accelerate convergence or to improve image quality (Xie et al., 2019; Kim et al., 2018; Gong et al., 2019);

(iii) direct reconstruction with deep learning alone without any use of traditional reconstruction methods (Whiteley and Gregor, 2019b; Zhu et al., 2018; Haeggstroem et al., 2018).

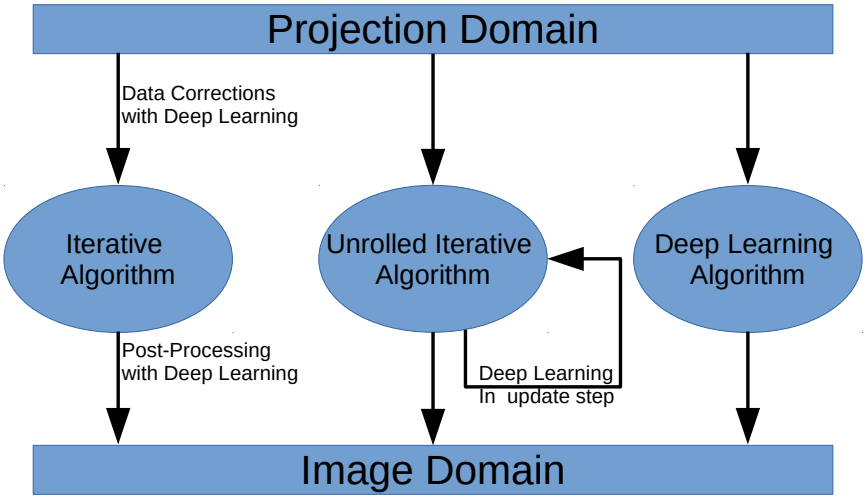


FIGURE 3.1: Deep Learning in Medical Image Reconstruction

Each of these categories are discussed along with reference to existing state of the art methods for CT and PET image reconstruction in this chapter.

3.1 Data Corrections or Post-processing

The use of deep learning for the development of either data corrections or post-reconstruction image based approaches has shown potential to improve the quality of reconstructed images. While it is possible to train a CNN to regress directly from the measurement (raw data) domain to the image domain, the use of CNN entirely in one particular domain makes it fast and relatively easy to implement. The motivation behind using deep learning architectures for these processing task is the extremely well documented performance in denoising and super resolution tasks. Data corrections involve improving the measurement data either though denoising or finding missing projection angle data. Post-processing in the image domain on the other hand consists of improving images reconstructed with standard reconstruction methods.

An example of data corrections by improving the raw data through scatter correction is proposed in Maier et al., 2018. In this work a modified U-Net is used to estimate scatter and correct the raw data in order to improve CT images. Sinogram repair is proposed by Whiteley and Gregor, 2019a, where a CNN is utilized to predict missing projection data for total body PET image reconstruction. The repaired sinograms eventually improve image reconstruction by standard methods. In CT imaging too missing projection data in sparse-view setting is estimated. An example in this regard is proposed in Lee et al., 2018, where the authors use U-Net to map sparse-view sinograms to full-view sinograms and then reconstruct the images using FBP.

Denoising the reconstructed PET images with a deep convolutional network was done in Gong et al., 2018. The authors used perceptual loss along with MSE to preserve qualitative and quantitative accuracy of the reconstructed images. The network was initially trained on simulated data and then fine-tuned on real patient data. The authors in Jin et al., 2017 use U-Net with a residual connection for denoising and artefact removal in the sparse-view estimate, while the work in Zhang et al., 2018 uses DenseNet with deconvolution for the same purpose. It is interesting to note that the networks have an encoder-decoder structure, wherein the encoder finds a compact representation of the input domain and the decoder learns to map this representation to the target domain. The dimensions of the input are reduced through the encoder as we go deeper into the layers. On the other hand, each of the decoder layers samples up these feature maps to eventually arrive at the output dimensions.

3.2 Unrolled Iterative Methods

Despite resulting in an improvement of the reconstructed output, the above mentioned methods do not directly intervene with the reconstruction process. This can be done using the two distinct frameworks (ii) and (iii). The first one involves the incorporation of a deep neural network into an unrolled iterative algorithm where a trained neural network accelerates the convergence by improving the intermediate estimates in the iterations (Gong et al., 2019; Xie et al., 2019; Kim et al., 2018). The hybrid methodology of unrolled iterative networks combines model-based and neural network approaches exploring the benefits of both methods. The paper by Gong et al. used a modified U-Net to represent images within the iterative reconstruction framework for PET images. The deep learning architecture was trained

on low-dose reconstructed images as input and high-dose reconstructed images as the output. The work by Xie et al. further extended this work by replacing the U-Net with a generative adversarial network (GAN) for image representation within the iterative framework. Kim et al incorporated a trained denoising convolutional neural network (DnCNN) along with a novel local linear fitting function into the iterative algorithm. The DnCNN which is trained on data with multiple noise levels improves the image estimate at each iteration. They used simulated and real patient data in their study. In Gupta et al., 2018, a U-Net is used to encode the prior, i.e., to project the current estimate to the prior image set while gradient descent enforces measurement consistency. Neural networks can be also used to replace traditional operators in optimization strategies as shown by Adler and Öktem, 2018. The reconstruction using these hybrid methods can be computationally expensive since it requires running an optimization procedure at test time.

3.3 Direct Reconstruction with Deep Learning

An alternative approach is using deep learning-based methods to directly map from projection to image space. Essentially neural network can be modeled to approximately learn the inverse mapping from measurement(y) to image (x). As represented in Fig 3.2, a neural network (F) with parameters (θ) can be represented as:

$$x = \boldsymbol{F}_{\hat{\theta}}\boldsymbol{y} \tag{3.1}$$

The challenge in this approach is the management of data and the number of parameters required for learning the mapping. Due to these challenges, this approach has been less explored compared to the two approaches discussed above.

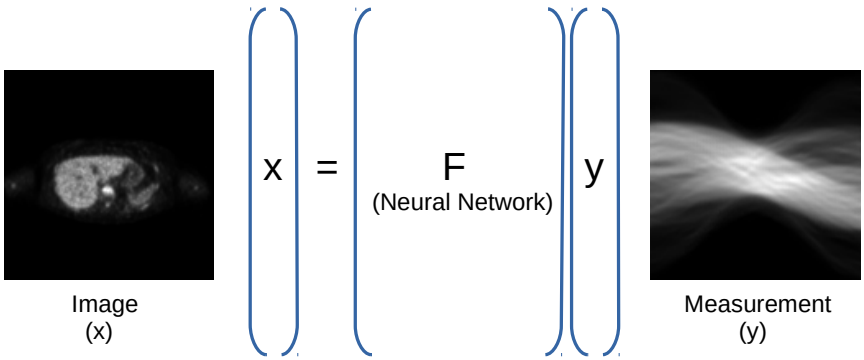


FIGURE 3.2: Direct image reconstruction with deep learning

The deep learning architecture proposed by Zhu et al. Zhu et al., 2018 called AUTOMAP uses fully connected (FC) layers (which encode the raw data information) followed with convolutional layers. The first three layers in this architecture are FC layers with dimensions $2n^2$,n^2 and n^2 respectively where $n \times n$ is the dimension of the input image. The AUTOMAP requires the estimation of a huge number of parameters which makes it computationally intensive. Although initially developed for magnetic resonance imaging (MRI), AUTOMAP has been claimed to work on other imaging modalities too. Brain images encoded into sensor-domain sampling strategies with varying levels of additive white Gaussian noise were reconstructed with AUTOMAP. Within the same concept of using FC layers' architectures a three stage image reconstruction pipeline called DirectPET has been proposed to reduce associated computational issues Whiteley and Gregor, 2019b. The first stage down-samples the sinogram data, following which a unique Radon transform layer encodes the transformation from sinogram to image space. Finally the estimated image is improved using a super resolution block. This work was applied to full body PET images and remains the only approach that can reconstruct multiple slices simultaneously (up to 16 images). DeepPET is another approach implemented on simulated images using encoder-decoder architecture based on the neural network proposed by the visual geometric group Haeggstroem et al., 2018. Using realistic simulated data, they demonstrated a network that could reconstruct images faster, and with an image quality (in terms of root mean squared error) comparable to that of conventional iterative reconstruction techniques.

In Li et al., 2019 the authors proposed an architecture termed iCT-Net consisting of 12 layers that are a combination of convolutions and modified fully-connected layers. The 12 layers are separated into segments and are trained separately before being combined for end-to-end training. To reduce the number of parameters in learning the mapping for full resolution CT reconstruction, Fu and De Man, 2019 proposed a breakdown of the problem into smaller fragments that can be mapped onto a hierarchical network architecture. The approach proposed in Ye et al., 2018 converts the sinogram data into a stack of back projections for each angle, which are then fed into a CNN. The spatial in-variance of the CNN is exploited to learn the mapping from these single view stacked back projections onto reconstructed images. Currently, we observe that adversarial networks are increasingly used in scenarios with high-resolution images. In Thaler et al., 2018 a Wasserstein generative adversarial network Arjovsky, Chintala, and Bottou, 2017 is proposed

for sparse-view CT image reconstruction. The authors used a combination of L_1 loss and adversarial loss to train their network. The generator in their work is a U-Net and the discriminator a typical classification CNN. It is to be noted that the authors performed their experiments on down-sampled images of resolution 128×128. Another methodology referred to as DUG-RECON Kandarpa et al., 2020, used a three-stage network to divide the image reconstruction problem into denoising, domain mapping and resolution improvement. They used a residual UNet for denoising the sinograms, then a double-UNet architecture to map the sinogram to image, and finally a super ResNet to improve image estimate. The approach was tested with both PET and CT data.

Chapter 4

DUG-RECON: A Framework for Direct Image Reconstruction using Convolutional Generative Networks

This paper explores convolutional generative networks as an alternative to iterative reconstruction algorithms in medical image reconstruction. The task of medical image reconstruction involves mapping of projection domain data collected from the detector to the image domain. This mapping is done typically through iterative reconstruction algorithms which are time consuming and computationally expensive. Trained deep learning networks provide faster outputs as proven in various tasks across computer vision. In this work we propose a direct reconstruction framework exclusively with deep learning architectures. The proposed framework consists of three segments, namely denoising, reconstruction and super resolution. The denoising and the super resolution segments act as processing steps. The reconstruction segment consists of a novel DUG which learns the sinogram-to-image transformation. This entire network was trained on PET and CT images. The reconstruction framework approximates 2-D mapping from projection domain to image domain. The architecture proposed in this proof-of-concept work is a novel approach to direct image reconstruction; further improvement is required to implement it in a clinical setting. In our work we explore the use of U-Net based deep learning architectures Ronneberger, Fischer, and Brox, 2015 to perform a direct reconstruction from the sinogram to the image domain using real patient datasets. Our aim is to reduce the number of trainable parameters along with exploring a novel strategy for direct image reconstruction using generative networks. More specifically our approach consists of a

three-stage deep-learning pipeline consisting of denoising, image reconstruction and super resolution segments. Our experiments included training the deep learning pipeline on PET and CT sinogram-image pairs. A single pass through the trained network transforms the noisy sinograms to reconstructed images. The reconstruction of both PET and CT datasets was considered and presented in the following sections.

4.1 Method

Image reconstruction with deep learning however is a data driven approach wherein there is a training and a prediction phase. Given a set of training data which is a subset of the raw data (y) and its corresponding images (x), a deep learning architecture learns the mapping from raw data to the image and improves this mapping through the training process. During the prediction phase a subset of raw data different from the training data serves as the input to the trained deep learning architecture. The output is a reconstructed image which is obtained on a single forward pass through the network. Hence making the reconstruction process through deep learning instantaneous as opposed to an iterative process. This makes direct reconstruction with deep learning faster and less computationally expensive than iterative algorithms.

4.1.1 Deep Learning Architectures

As shown in Figure 4.1, we propose a three-stage deep learning pipeline for the task of tomographic reconstruction. In the first step the raw data (projection space) are denoised. Next the denoised sinograms are transformed to the image domain in the image reconstruction segment. The third and final segment operates in the image domain to improve the image produced after domain transformation. The following sections discuss these segments in detail.

FIGURE 4.1: Proposed Deep Learning pipeline for Direct Image Reconstruction

TABLE 4.1: Trainable Parameters comparison

Architecture	Input Size	Output Size	Trainable Parameters
AUTOMAP	200×168	200×200	6,545,920,000
Radon Inversion Layer (40×40 Patch size)	200×168	200×200	382,259,200
DeepPET	128×128	128×128	62,821,473
DUG-RECON	128×128	128×128	17,444,140

Denoising

We used a modified U-Net architecture to denoise the Poisson sampled sinograms, based on the work previously carried out for ultrasound denoising Perdios et al., 2018. The U-Net is an encoder-decoder network which was initially implemented for segmentation but over the years its applications have broadened. As shown in Figure 4.2 there are increasing number of convolutions along with max pooling to arrive at an encoding of the input and then with convolutions followed by upsampling, arriving at the output with an identical dimension as the input. The important modification in the architecture mentioned in Perdios et al., 2018 with respect to U-Net was the residual connection from the input to the final output. Perdios et al trained the denoising architecture on simulated ultrasound images so as to enhance ultrafast ultrasound imaging. This denoising architecture corresponds to the first segment in our proposed framework. It was trained on raw data pairs, i.e., low-count and high-count sinograms, considering multiple noise levels. The detailed architecture is represented in Figure 4.2. We defined the loss function between a true sinogram $\boldsymbol{y}^\star = [y_1^\star, \ldots, y_n^\star]^\top \in \mathbb{R}^n$ and a prediction $\hat{\boldsymbol{y}} = [\hat{y}_1, \ldots, \hat{y}_n]^\top \in \mathbb{R}^n$ as the MSE:

$$\mathrm{MSE}(\boldsymbol{y}^\star, \hat{\boldsymbol{y}}) = \frac{1}{n} \sum_{i=1}^{n} (y_i^\star - \hat{y}_i)^2 , \tag{4.1}$$

where, n is the number pixels on the sinogram, corresponding to the number of detectors in the scanner.

Image Reconstruction

The novelty in this work is the proposed U-Net based network in contrast to previous works in direct image reconstruction using the FC layer architectures. This design of the network draws its inspiration from conditional GAN for image to image translation called Pix2Pix Isola et al., 2017. The

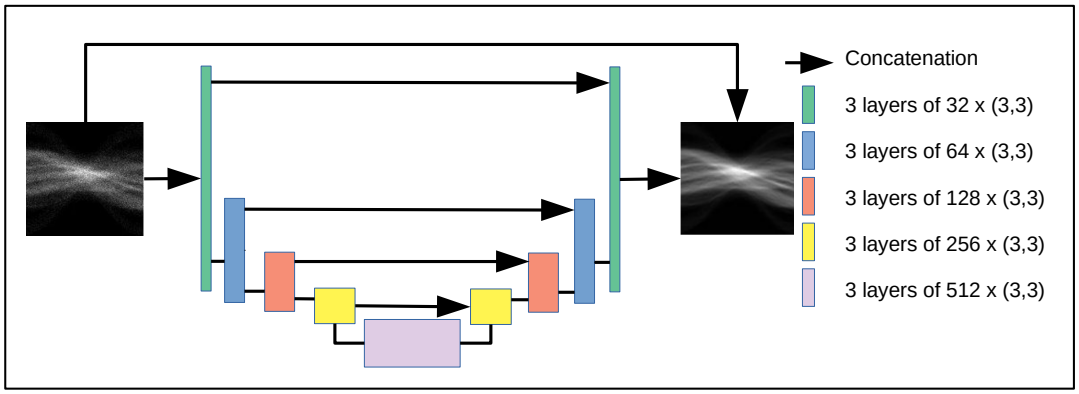


FIGURE 4.2: Representation of the denoising network. The inputs to the network were 2-D grayscale slices with resolution 128×128 and the outputs were denoised sinograms.

proposed network namely DUG consists of two cascaded U-Nets. The first U-Net transforms the raw data to image while the second U-Net takes as input the generated image and transforms it back to the raw data. The second U-Net assesses the reconstructed image output, reiterating the relation between the sinogram and the image. This architecture differs from the Pix2Pix, which consists of a generator (U-Net like network) and a discriminator (classification convolutional network). While the generator in both architectures serves the purpose of transforming images from one domain to the other, the discriminator with regards to Pix2Pix classifies inputs as real/fake. The objective function for this architecture can be written as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{G_1} + \mathcal{L}_{G_2} + \mathcal{L}_{G_1+G_2} \tag{4.2}$$

where,

$$\mathcal{L}_{G_1} = \frac{1}{n}\sum_{i=1}^{n} |x_i^\star - \hat{x}_i| \tag{4.3}$$

$$\mathcal{L}_{G_2} = \frac{1}{n}\sum_{i=1}^{n} |y_i^\star - \hat{y}_i| \tag{4.4}$$

G_1, G_2 are Generator 1 and Generator 2 which predict image and sinogram respectively; $x^\star$, $\hat{x}$ the true and predicted image, $y^\star$, $\hat{y}$ the true and predicted sinogram respectively. $\mathcal{L}_{G_1+G_2}$ is defined similar to $\mathcal{L}_{G_2}$ with the combined architecture of $G_1 + G_2$, keeping the weights of G2 fixed.

The architecture is represented in detail in Figure 4.3. The training for this architecture is summarized in Algorithm 1. A comparison of the trainable parameters for various segments that are used to perform the domain mapping from sinogram to image is provided in Table 5.1, considering the

AUTOMAP from Zhu et al., 2018, the Radon inversion layer from Whiteley and Gregor, 2019b and the proposed architecture DUG along with denoising and super resolution segments.

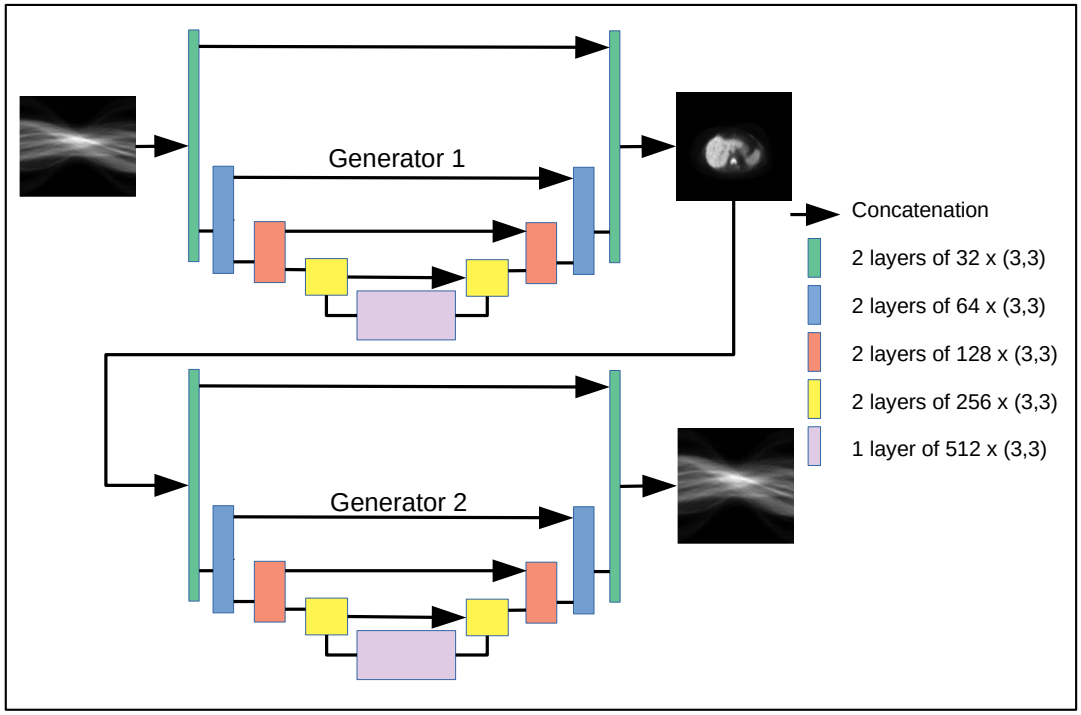


FIGURE 4.3: Representation of the DUG, the image reconstruction block. This network was trained on denoised sinograms which were the outputs of the previous segment.

Super Resolution

The function of the SR is to improve the estimate produced by the image reconstruction network. Several works already exist concerning single image super resolution Ledig et al., 2017; Lim et al., 2017. In this work we employed a basic super residual network architecture to improve the reconstruction. It consists of convolutional blocks followed by batch normalization with parametric rectified linear unit (PReLU) activation. There were a total of 8 residual blocks in the network as represented in Figure 4.4. The loss function used in this architecture was perceptual loss:

$$\text{PerceptualLoss} = |\text{VGG}_{16}(\boldsymbol{x}^{\star}) - \text{VGG}_{16}(\hat{\boldsymbol{x}})| \tag{4.5}$$

$\text{VGG}_{16}(\boldsymbol{x}^{\star})$ and $\text{VGG}_{16}(\hat{\boldsymbol{x}})$ are the extracted features with VGG_{16} convolutional neural network (Simonyan and Zisserman, 2014) for the true and predicted image.

Algorithm 5: Training the DUG

M = number of epochs ;
N = total training data (images/sinograms) ;
for i = 1,2,. . . , M **do**
  for j = 1,2,. . . ,N **do**
    Train G_1: minimizing $\mathcal{L}_{G_1}$;
  end
  for j = 1,2,. . . ,N **do**
    Train G_2: minimizing $\mathcal{L}_{G_2}$;
  end
  for j = 1,2,. . . ,N **do**
    Train combined architecture, freezing the weights of G_2:
     minimizing $\mathcal{L}_{G_1+G_2}$;
  end
end

The features are extracted from the 10th layer of the VGG architecture i.e., only the first three convolutional blocks are considered. We observed that extracting deeper features led to the network hallucinating features in the reconstructed images.

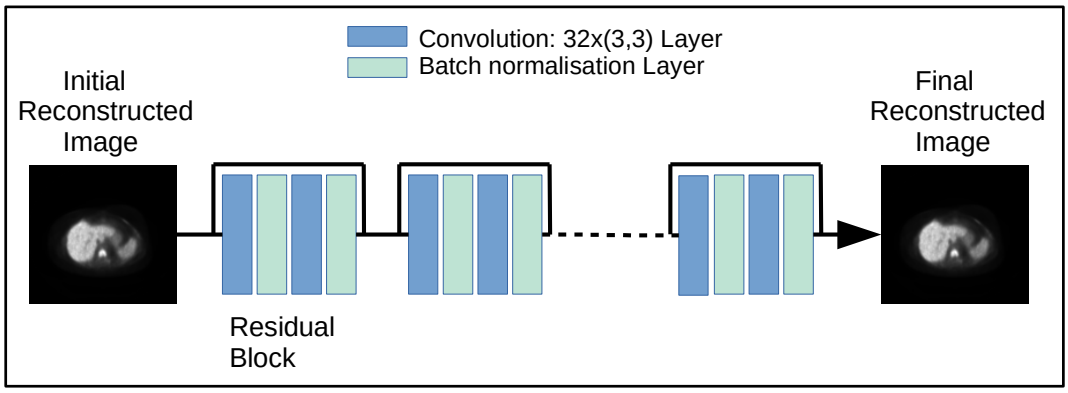


FIGURE 4.4: Representation of the super resolution block. It consists of 8 residual blocks with Convolution, Batch normalization and PReLu.

4.2 Dataset Description

We applied our methodology on fluorothymidine (FLT) PET/CT images from the American College of Radiology Imaging Network (ACRIN) FLT Breast PET/CT database Kostakoglu et al., 2015. The details of the dataset are given in Table 5.2. The sinograms were initially generated by projecting 2-D PET

and CT images slices with the Python SKLEARN Radon transform, following the models (1.16) and (1.13) for CT and PET respectively, with Poisson noise added. The methodology represented in Figure 4.5 was used for data preparation for the PET and CT modalities respectively. Sample pairs from the PET and CT datasets are shown in Figures 4.6 and 4.7. The CT images were downsized from 512×512, and the reconstruction was implemented for 2-D 128×128 images.

TABLE 4.2: Dataset Description

Dataset Statistics	
Modalities	CT, PET
Number of Patients	83
Number of PET 2-D Image slices	76,000
Number of CT 2-D Image slices	21,104
PET Matrix size	128
CT Matrix size	512
Scanner	GE Discovery ST

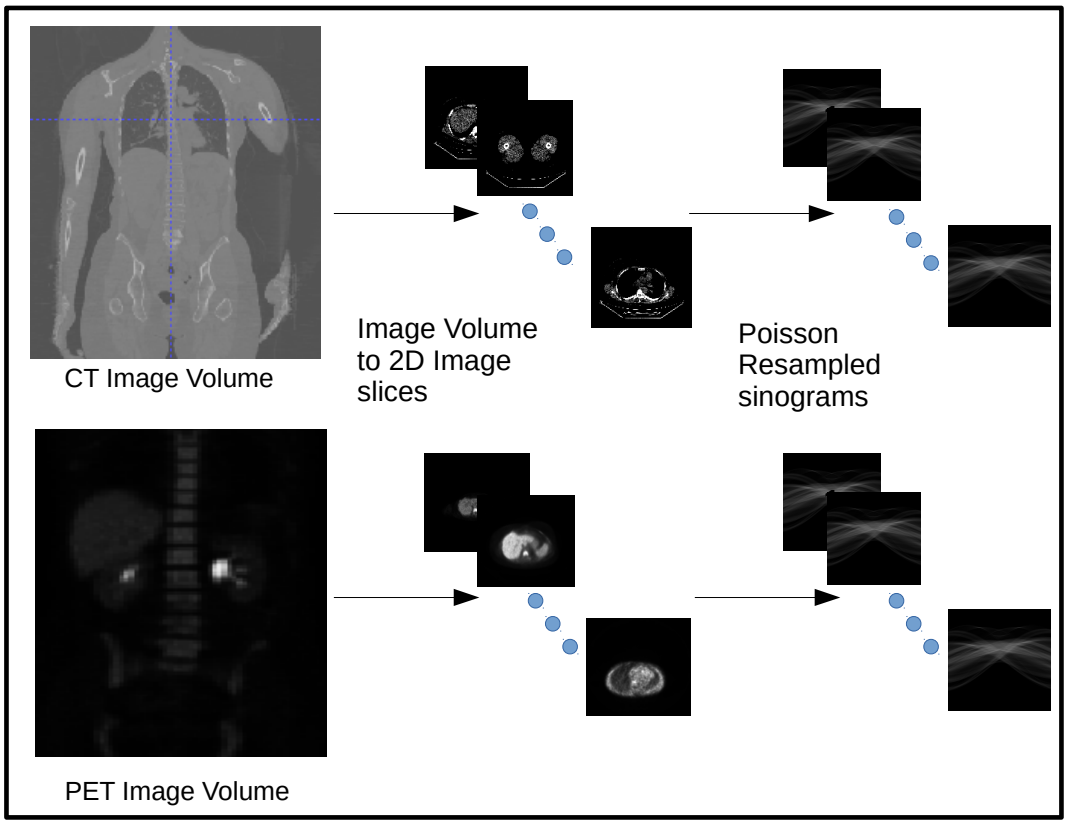


FIGURE 4.5: Data preparation

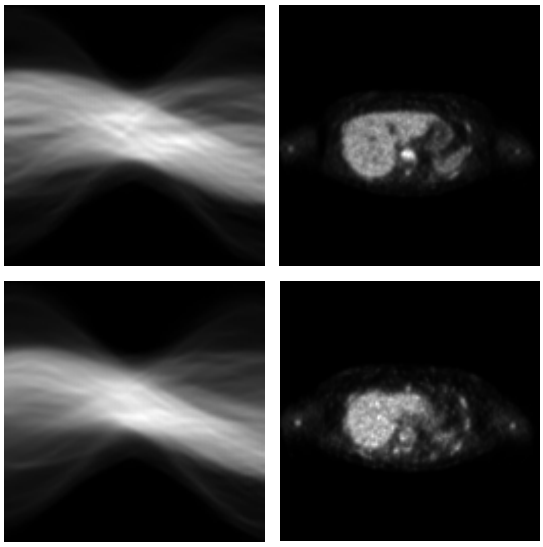


FIGURE 4.6: Example PET sinogram-image pairs from the dataset

4.3 Training

TensorFlow Abadi et al., 2016 and Keras Chollet, 2015 were used for the realization of the architectures described in the section above. These architectures were implemented on a single Nvidia GeForce GTX 2080Ti GPU. A collection of images $\{\boldsymbol{x}_k^\star\}_{k=1}^N$ was used to generate a corresponding collection of noiseless sinograms $\{\boldsymbol{y}_k^\star\}_{k=1}^N$ following models (1.16) and (1.13), low-counts and high-counts sinograms, $\{\boldsymbol{y}_k^{\text{LC}}\}_{k=1}^N$ and $\{\boldsymbol{y}_k^{\text{HC}}\}_{k=1}^N$ sinograms were generated by adding Poisson noise, where the expected number of counts was adjusted by rescaling the intensity. The denoising segment was trained with the collection $\{(\boldsymbol{y}_k^{\text{LC}}, (\boldsymbol{y}_k^{\text{HC}})\}_{k=1}^N$, using the high-count sinograms as ground truth, with total number of training samples $N = 120000$ and number of epochs $M = 100$. For the training of the second segment, we used a collection of denoised sinograms, namely $\{\hat{\boldsymbol{y}}_k\}_{k=1}^N$, and their corresponding ground true images $\{\boldsymbol{x}_k^\star\}_{k=1}^N$. The image reconstruction segment was trained according to the Algorithm 5, that is to say by alternating between training G_1 with $\{(\hat{\boldsymbol{y}}_k, \boldsymbol{x}_k^\star)\}_{k=1}^N$ and training G_2 with $\{(\hat{\boldsymbol{x}}_k, \boldsymbol{y}_k^\star)\}_{k=1}^N$, where $\hat{\boldsymbol{x}}_k$ is a prediction from G_1 with $\hat{\boldsymbol{x}}_k$ as an input. This segment was trained with $N = 40000$ and $M = 50$. The CT data were augmented by rotating the data by 90 degrees to generate the required training data. Owing to the larger PET dataset it was not necessary to perform data augmentation. Finally the SR segment was trained on the images predicted by the DUG and the GT images $\{(\hat{\boldsymbol{x}}_k, \boldsymbol{x}_k^\star)\}_{k=1}^N$. The SR block was trained with $N = 20000$ for 100 epochs. For the testing, a set of 2000 sinogram-image pairs were used.

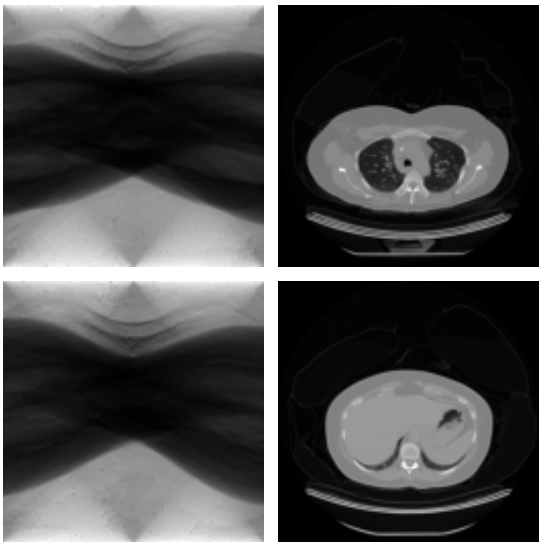


FIGURE 4.7: Example CT sinogram-image pairs from the dataset

4.4 Quantitative analysis

Testing for the aforementioned architectures was done on samples that were not a part of the training data. The metrics used for this analysis are RMSE and SSIM Index. They are defined below:

$$\text{RMSE}(\boldsymbol{x}^\star, \hat{\boldsymbol{x}}) = \sqrt{\frac{1}{n}\sum_{j=1}^{m}(x_j^\star - \hat{x}_j)^2} \tag{4.6}$$

where n is the number of pixels. $\hat{x}$ is the GT $x^\star$ the predicted output.

$$\text{SSIM}(\boldsymbol{x}^\star, \boldsymbol{x}) = \frac{(2\mu_{x^\star}\mu_x + c_1)(2\sigma_{x^\star x} + c_2)}{(\mu_{x^\star}^2 + \mu_x^2 + c_1)(\sigma_{x^\star}^2 + \sigma_x^2 + c_2)} \tag{4.7}$$

where $\mu_{x^\star}$ μ_x are the averages of $\boldsymbol{x}^\star$ and $\boldsymbol{x}$ respectively, $\sigma_{x^\star}^2$ and σ_x^2 are the variances of $\boldsymbol{x}^\star$ and $\boldsymbol{x}$, $\sigma_{x^\star x}$ is the covariance between $\boldsymbol{x}^\star$ and $\boldsymbol{x}$, $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$ where $k_1 = 0.01$ and $k_2 = 0.03$ by default.

4.4.1 Region of Interest analysis

The SNR and CNR were studied for four regions of interest identified within the patient body. The SNR and CNR were evaluated by treating a region as foreground and the other three regions as background.

$$\text{SNR} = \frac{\mu_\text{r} - \mu_\text{b}}{\sigma_\text{b}} \tag{4.8}$$

$$\text{CNR} = \frac{|\mu_r - \mu_b|}{\sqrt{\sigma_r^2 + \sigma_b^2}}. \tag{4.9}$$

where μ_r and μ_b, σ_r and σ_b correspond to the mean and standard deviation in the region of interest (ROI) and the background respectively. In this study we compared the initial reconstructed output of the DUG, the final reconstruction along with SR and the original GT which was reconstructed with GE discovery ST using an OSEM algorithm..

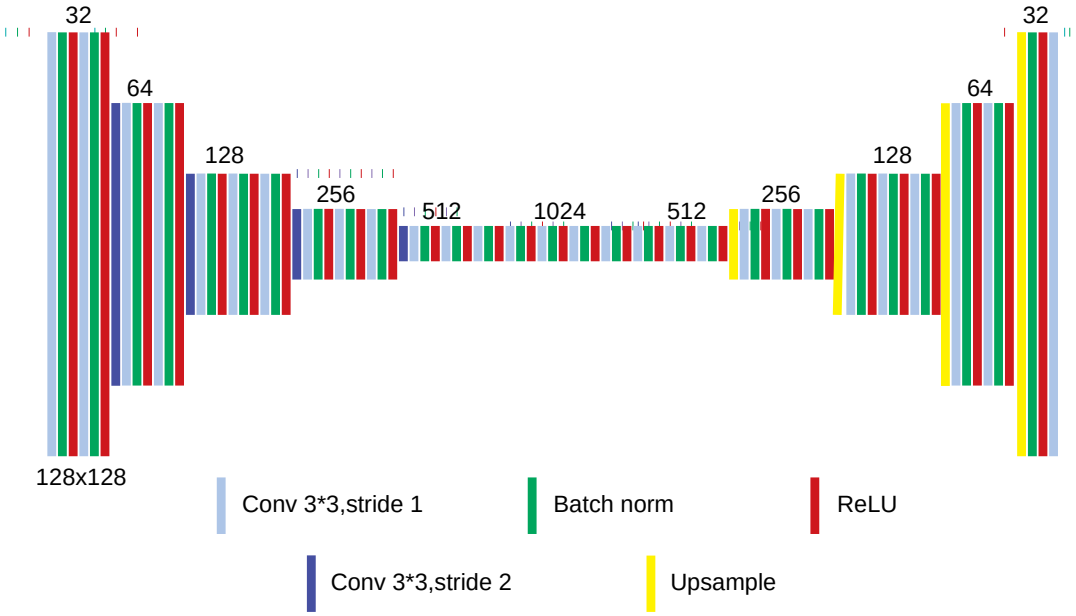


FIGURE 4.8: Representation of DeepPET. The number of filters in each convolutional layer is labeled on top of each block.

4.5 Comparison with DeepPET

We implemented the architecture DeepPET Haeggstroem et al., 2018 and compared the predictions with our proposed approach for the reconstruction of PET images. DeepPET was trained on $\{(\hat{\boldsymbol{y}}_k, \boldsymbol{x}_k^\star)\}_{k=1}^N$, notation similar to the training section from above, with $N = 120000$, exclusively on PET data. It is worth noting that the input and output dimensions in our study are identical while it was not originally for DeepPET. The architecture of DeepPET is summarized in Figure 4.4. This architecture was trained for 100 epochs with an Adam optimiser.

4.6 Results

The predictions from the architectures along with the GT and the sinogram are shown in Figure 4.9 for PET images. The results are displayed for four test image slices across the columns. Each column shows the predicted output by the proposed DUG-RECON architecture and the DeepPET architecture, as well as the GT. With regards to the proposed architecture it is observed that the initial reconstructed image i.e., the output of the DUG looks blurred while final reconstructed output from the super resolution block has noticeably improved details. The predictions by DeepPET are also visibly blurred compared to the final reconstructed output of the proposed architecture and the GT. These observations are further ascertained in Table 5.3 where the quantitative metrics are tabulated. Figure 4.13 provides a comparison of the intensity profiles for the predictions by DUG, DUG+SR and DeepPET w.r.t. to the GT for PET images. These intensity values are observed along the line marked in yellow in these figures. As this figure shows, the intensity profile of the final reconstructed image of the proposed architecture is closest to the GT. The predictions by DUG and DeepPET are smoother compared to the predictions by DUG+SR and the GT.

TABLE 4.3: The SSIM and RMSE for the various modalities compared

Image	Architecture	RMSE	SSIM
1	DUG	0.059	0.74
	DUG+SR	0.038	0.84
	DeepPET	0.047	0.80
2	DUG	0.043	0.76
	DUG+SR	0.046	0.86
	DeepPET	0.054	0.85
3	DUG	0.050	0.76
	DUG+SR	0.038	0.85
	DeepPET	0.043	0.83
4	DUG	0.061	0.70
	DUG+SR	0.045	0.82
	DeepPET	0.048	0.79

The ROI analysis is tabulated in Table 5.5 for the four regions marked in Figure 4.10. This analysis was carried out for final predictions by the proposed architecture and MLEM. Looking closely at Table 5.5 we notice that the mean values of the deep learning predicted image and the MLEM reconstructed image are comparable. The results for CT images are displayed in

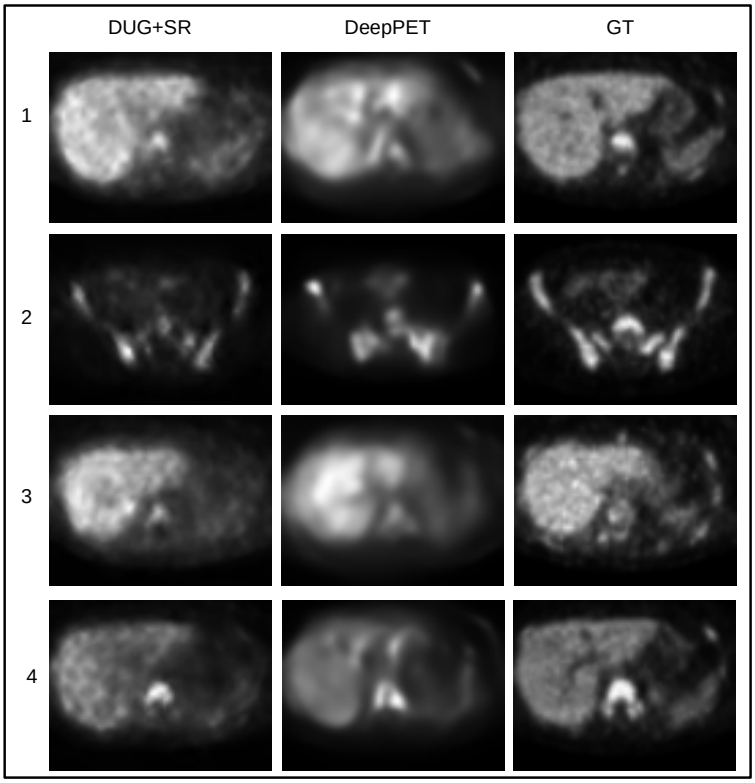


FIGURE 4.9: Image predictions by DUG+SR, DeepPET and GT for four PET Images from different parts of the patient volume

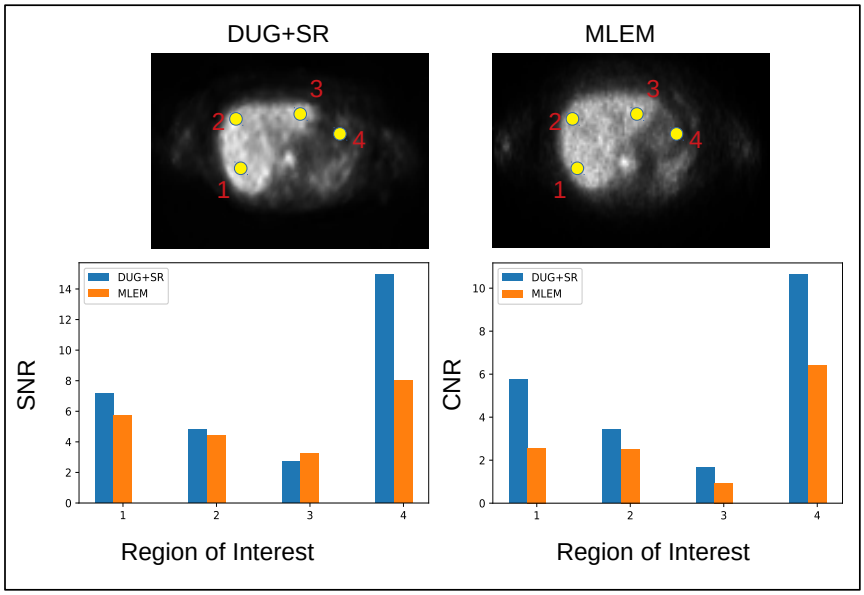


FIGURE 4.10: SNR and CNR comparison amongst DUG+SR and OSEM for PET image along 4 regions of interest

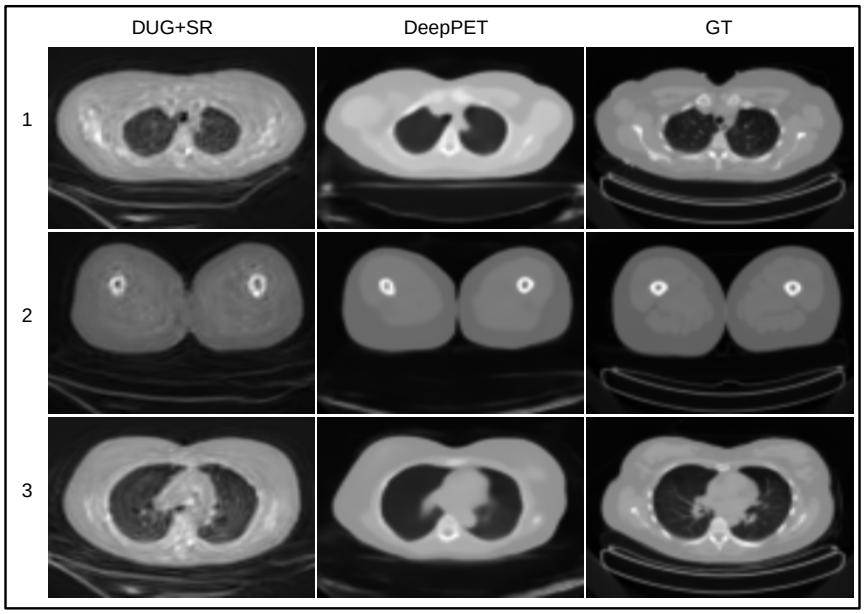


FIGURE 4.11: Image predictions by DUG+ SR, DeepPET and GT are displayed for 3 CT Images along different parts of the patient volume.

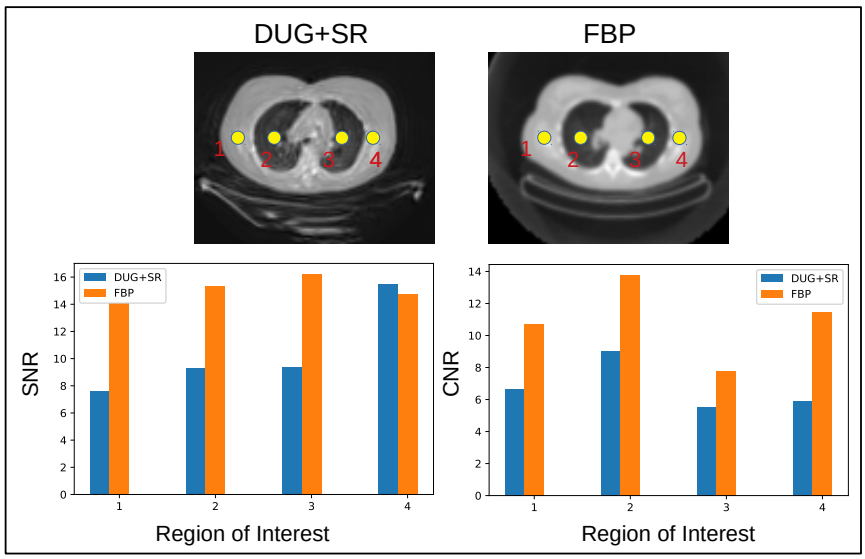


FIGURE 4.12: SNR and CNR comparison amongst DUG+SR and OSEM for CT image along 4 regions of interest

TABLE 4.4: The SSIM and RMSE for the CT images are evaluated for 4 different 2-D slices. Here the architecture indicates the prediction by DUG and that of DUG along with SR segment

Image	Architecture	RMSE	SSIM
1	DUG	0.0083	0.90
	DUG+SR	0.0015	0.98
	DeepPET	0.0012	0.99
2	DUG	0.0081	0.90
	DUG+SR	0.0015	0.99
	DeepPET	0.0014	0.99
3	DUG	0.0015	0.91
	DUG+SR	0.0018	0.98
	DeepPET	0.0013	0.99

Figure 4.11. This Figure provides a comparison between reconstructed image predictions with the proposed architecture, DeepPET with respect to the GT. The high-resolution nature of the CT images and a smaller dataset presented challenges during the training of the proposed architecture.

TABLE 4.5: ROI Analysis: The mean, SD and the SNR for the 4 regions of interest marked in Figure 12

Region	Image	Mean	SD	SNR	CNR
1	DUG+SR	0.706	0.024	7.15	5.71
	MLEM	0.676	0.035	5.72	4.55
2	DUG+SR	0.713	0.091	4.81	3.42
	MLEM	0.648	0.11	4.38	3.26
3	DUG+SR	0.744	0.071	2.73	1.65
	MLEM	0.547	0.154	3.23	1.22
4	DUG+SR	0.117	0.008	14.96	10.64
	MLEM	0.057	0.010	8.01	4.8

TABLE 4.6: ROI Analysis: The mean, SD and the SNR for the 4 regions of interest marked in Figure 15

Region	Image	Mean	SD	SNR	CNR
1	DUG+SR	0.011	2.91e-4	7.61	6.66
	FBP	0.011	3.45e-4	15.86	10.69
2	DUG+SR	0.004	1.53e-4	9.34	9.02
	FBP	0.004	1.84e-4	15.36	13.74
3	DUG+SR	0.005	5.90e-4	9.40	5.49
	FBP	0.005	4.88e-4	16.19	7.78
4	DUG+SR	0.012	8.319e-4	15.47	5.92
	FBP	0.012	2.736e-4	14.74	11.47

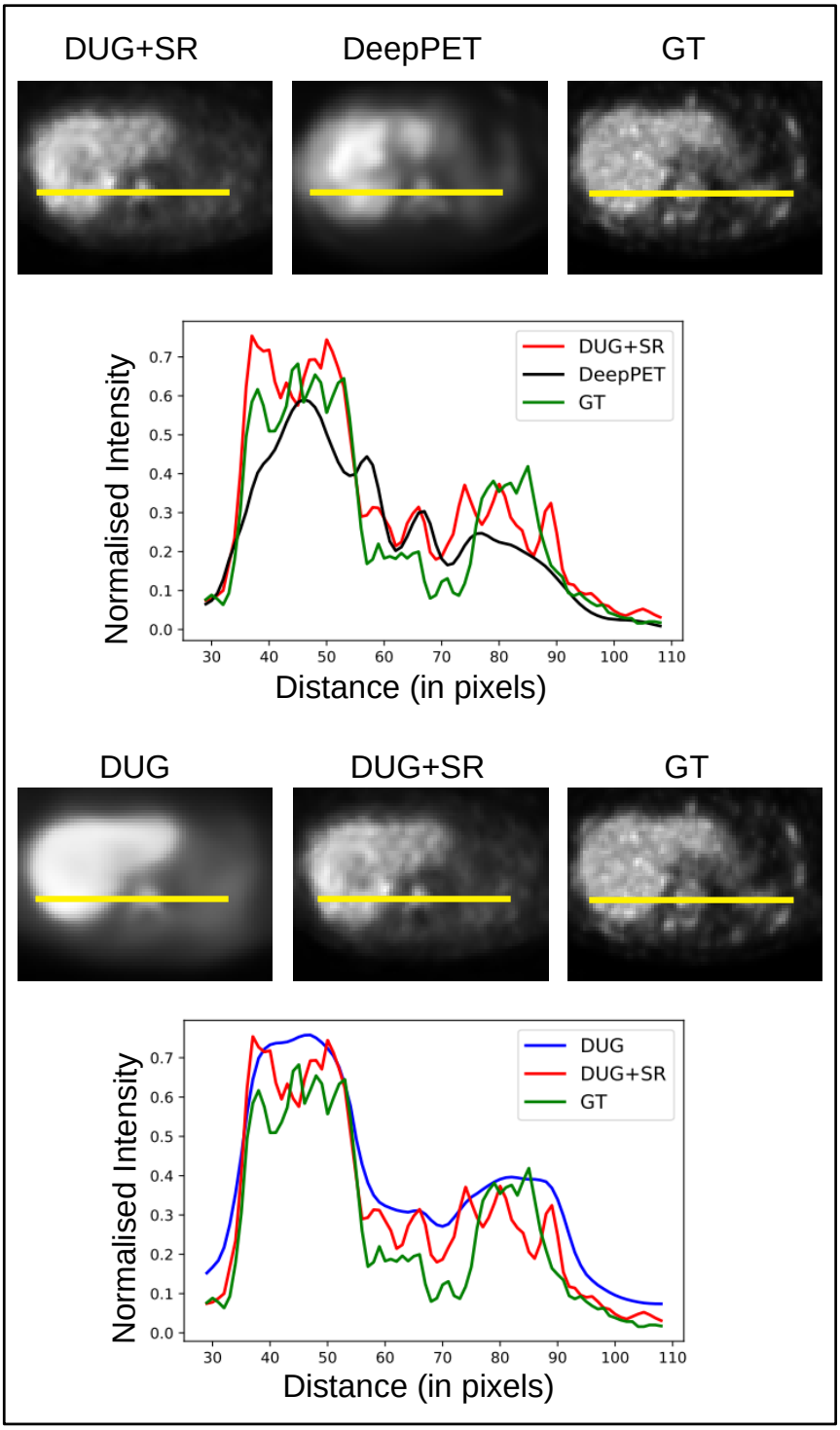


FIGURE 4.13: Intensity Profile across the image (highlighted by a yellow line) for a PET image prediction by DUG, DUG+SR and DeepPET compared with the GT

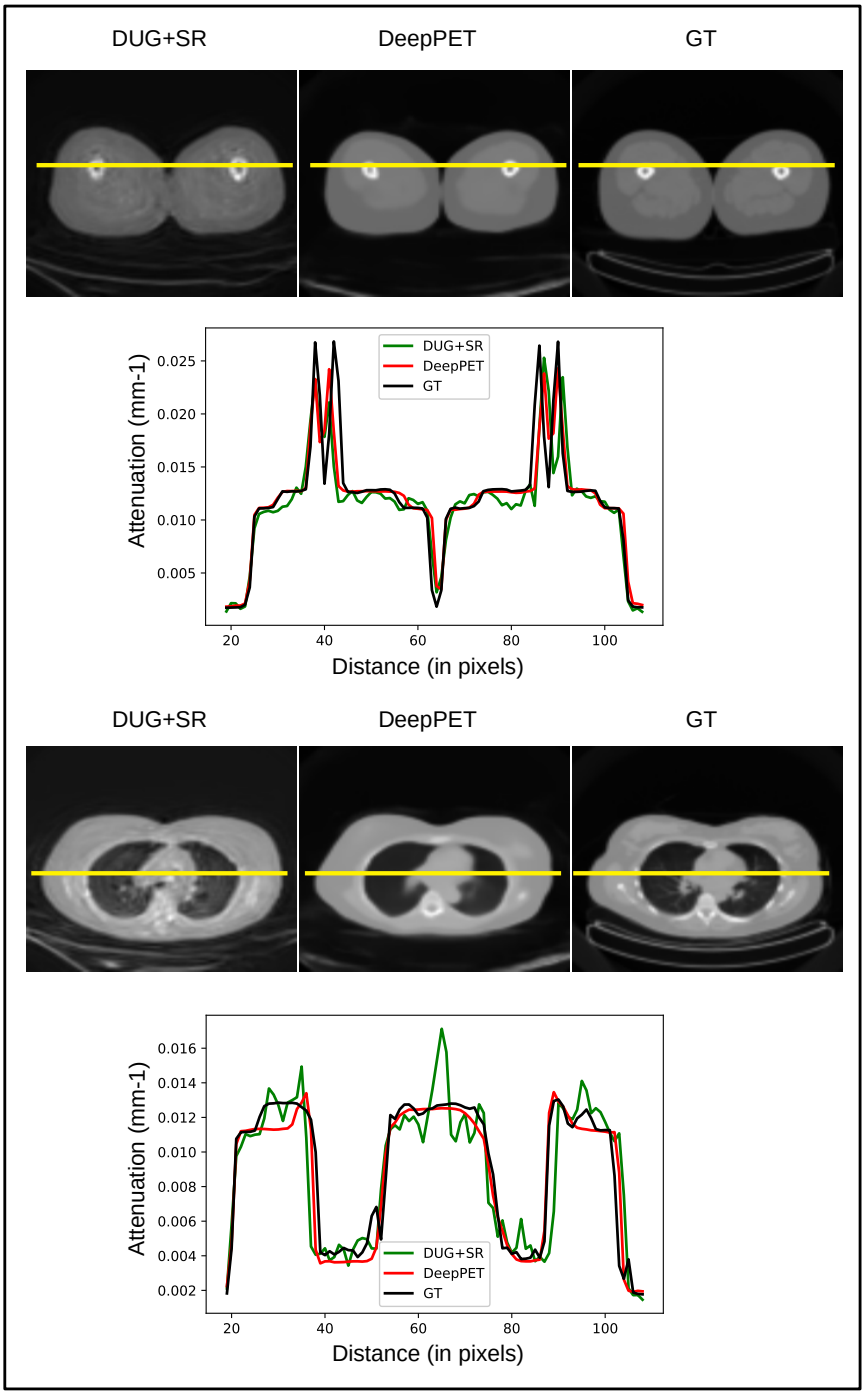


FIGURE 4.14: Intensity Profile for two CT images (highlighted by a yellow line) predicted by DUG and SR compared with the GT

The predictions by DeepPET appear to be better resolved than those by the proposed architecture. However, the tissue and the bone structures are not clearly seen in the predictions by the deep learning architectures, thereby requiring further work to improve the reconstruction. The intensity plots are compared for two different images in the Figure 4.14. The ROI analysis was carried out for 4 regions marked in the images reconstructed with deep learning and FBP. The image reconstructed with FBP has better SNR and CNR compared to the image reconstructed with the proposed architecture.

4.7 Discussion

Deep learning has been applied to different fields of medical imaging. The vast majority of developments concern primarily image processing and analysis/classification tasks. Few works devoted in the field of image reconstruction have been largely concentrated in the use of deep learning within classical tomographic reconstruction algorithms. The main objectives of these works have been an improvement in the speed of convergence and the quality of the successive image estimation within the iterative reconstruction process. The alternative approach involving direct image reconstruction through the use of deep learning approaches to estimate images directly from the use of raw data (sinograms or projections) has been much less explored both for PET and CT.

Most implementations in direct image reconstruction concern the use of fully connected layers which encode the raw data followed by convolutional layers. In most of the proposed implementations a large number of parameters need to be optimised which reduces the computational burden and overall robustness. In this work we have proposed an original direct image reconstruction deep learning framework based on an architecture inspired by convolutional generative adversarial networks used in image to image translation. The implementation is based on the use of a double U-Net generator (DUG) consisting of two cascaded U-Nets. While the first network transforms the raw data to an image the second one assesses the reconstructed image output of the first network by reiterating the relationship between the reconstructed image and the raw data. Two additional blocks were added; namely a network denoising the raw data prior to their input in the DUG network and a super-resolution block operating on the DUG output image in order to improve it's overall quality. The proposed network was directly

trained on clinical datasets for both PET and CT image reconstruction and its performance was assessed qualitatively and quantitatively.

Deep neural networks usually result in blurred output. This fact is clear in the predictions made by the DUG network. Both the qualitative analysis using the profiles through the reconstructed images and the quantitative metrics SSIM and the MSE, demonstrate the improvement of the reconstructed images resulting from the incorporation of the SR block. The qualitative analysis also clearly demonstrates the superiority of the proposed algorithm for direct PET image reconstruction in comparison to alternative approaches such as DeepPET. Finally in the ROI analysis we observed that the SNR and CNR are higher with the deep learning approach for the PET images while they are lower than the traditional methods for CT images. This is consistent with the observations in the qualitative analysis, where the proposed approach was not able to sufficiently resolve different tissue classes in the resulting reconstructed CT images in comparison with the ground truth.

One of the potential reasons of the worse performance of DUG-RECON for CT reconstruction relative to the superior performance observed for PET image reconstruction may be the lower number of available CT images in the training process. This limitation will be addressed as part of future work. Despite the lower performance of the proposed architecture for CT images it still presents comparable predictions and opens up avenues for deep learning architectures in tomographic reconstruction. In general, the limitations of a deep learning based reconstruction is the adaptability to new data which is very different from the training sample space. Once a practical methodology is identified, one could have a deep learning pipeline with an ensemble of networks trained on different datasets to perform the reconstruction task.

4.8 Conclusion

We have demonstrated the use of generative convolutional networks for the tomographic image reconstruction task. More specifically we have proposed a new architecture for direct reconstruction that approximates the 2-D reconstruction process. Also we have significantly reduced the parameters required for the domain transform task in image reconstruction. The three-step training pipeline based exclusively on deep learning decentralises the various tasks involved in image reconstruction into denoising, domain transform and super resolution. Various super resolution strategies are currently being explored to improve the reconstructed image. Our proposed strategy for

tomographic reconstruction will eventually lead to a network based reconstruction as we continue to improve the framework. Currently it does not perform better than traditional methods in terms of utility metrics but still has the advantage of instantaneous reconstruction and an effective denoising strategy. We plan to extend the work on realistic detector data generated through Monte Carlo simulations in addition to sinograms obtained through Radon transform. We are also working on adapting the architecture to raw detector data. Another important aspect of the data based deep learning approach is that the predictions are limited by the quality of the dataset. It becomes essential to have realistic datasets without compromising on the image quality to improve the training of the neural networks.

Chapter 5

LRR-CED: Low-Resolution Reconstruction aware Convolutional Encoder-Decoder Network for Direct Sparse-View CT Image Reconstruction

Sparse-view CT reconstruction has been at the forefront of research in medical imaging. Reducing the total X-ray radiation dose to the patient while preserving the reconstruction accuracy is a big challenge. The sparse-view approach is based on reducing the number of rotation angles, which leads to poor quality reconstructed images as it introduces several artifacts. These artifacts are more clearly visible in traditional reconstruction methods like the FBP algorithm. Over the years, several model-based iterative and more recently deep learning-based methods have been proposed to improve sparse-view CT reconstruction. Many deep learning-based methods improve FBP-reconstructed images as a post-processing step. In this work, we propose a direct deep learning-based reconstruction that exploits the information from low-dimensional FBP estimates, to learn the projection-to-image mapping. This is done by concatenating the FBP estimate at multiple resolutions in the decoder part of a CED. This approach is investigated on two different networks, based on Dense Blocks and U-Net to show that a direct mapping can be learned from a sinogram to an image. The results are compared to a post-processing deep learning method and an iterative method that uses a TV regularization.

5.1 Main Contribution

The main drawbacks of current deep learning-based direct image reconstruction algorithms are the tedious training process necessary to train large networks with large number of trainable parameters and the requirement of high memory in case of high-resolution CT images. In this work we propose a new method for direct deep learning based sparse-view CT image reconstruction with fully convolutional networks. We use two networks, namely Fully Convolutional Densenets Jégou et al., 2017 and U-Net Ronneberger, Fischer, and Brox, 2015. An important characteristic of both these architectures Jégou et al., 2017; Ronneberger, Fischer, and Brox, 2015 is the presence of concatenation from the encoding layers to the decoding layers that ensures the usage of features from the input for the reconstruction. Specifically, for application in sparse-CT image reconstruction, the network would have sparse-view sinograms as input and reconstructed images as output. The original application in the medical imaging field of both these architectures was in image segmentation, where the image-to-image mapping operates in the same image domain. Medical image reconstruction on the other hand involves mapping between two different domains (sinogram to image). In order to help the network to learn the mapping from sinogram to image, we propose the use of FBP image estimates of the sparse sinograms and concatenate them with the feature maps of the decoder.

Given that we only have access to sparse measurement data, taking the form of a sinogram $\boldsymbol{y}$, we can enforce that the inverse mapping $\boldsymbol{F}$ at each layer/sub-resolution of the network is consistent in the measurement domain. That is $\boldsymbol{PF}(\boldsymbol{y}) = \boldsymbol{y}$. This can be achieved by concatenating, as feature maps, (fast) low-resolution FBP-reconstructed images for each or a subset of the network levels. While this leads to a massive reduction of the parameters (fully convolutional layers instead of fully-connected) required in the network, the above-mentioned constraint is not enough to learn the inverse mapping as it cannot capture information about the image $\boldsymbol{x}$ outside the range of the physical under-determined operator $\boldsymbol{P}$ (Radon transform for CT). Hence, the network needs to be trained accordingly.

Once the network is trained, these custom concatenations enable architectures that were previously used for denoising/artifact removal to learn a mapping from sparse sinograms to full-resolution CT images. One characteristic feature of reconstructions generated by deep learning-based methods is the blurriness of the outputs. To counteract this we used perceptual

loss involving features extracted from two different levels of VGG16 network (Block 1 and Block 3). Since the exclusive use of perceptual loss results in unrealistic artifacts we couple it with a L_1 loss. A general representation of the proposed approach is depicted in Figure 5.1. It consists of a CED network with two blocks in both the encoder and the decoder that takes in as input a reshaped sparse sinogram which has the same dimensions as the output image. A concatenation of two resolutions $h_1 \times w_1$ and $h_2 \times w_2$ is incorporated in the decoder.

The main contributions of our work are summarized as follows:

- A new approach for sparse-view CT image reconstruction using fully-convolutional networks
- Use of lower resolution FBP estimates which enable the networks that are predominantly used for denoising to learn the more complex mapping from sinogram to image domain.
- Two neural networks are implemented to test this approach using different levels of sparsity in the sinograms.

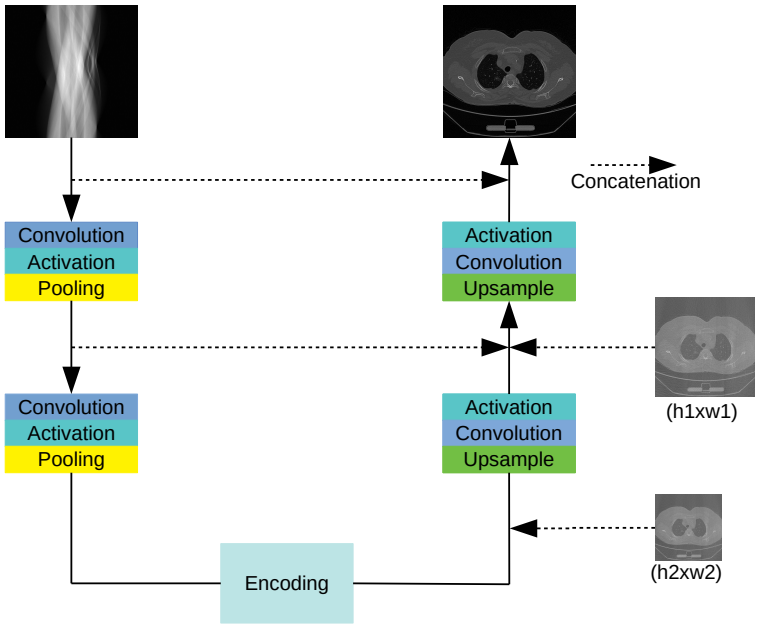


FIGURE 5.1: General representation of an encoder-decoder architecture with fully convolutional layers and the proposed FBP concatenations (x_1 and x_2) at two different resolutions $h_1 \times w_1$ and $h_2 \times w_2$

5.2 Methods

5.2.1 Proposed Low Resolution Reconstruction aware CED Model

Supervised deep learning-based methods learn the mapping between the measurement $\boldsymbol{y}$ and the corresponding reconstructed image $\boldsymbol{x}$. In the case of direct deep learning-based image reconstruction this mapping is typically learned via neural networks which can be represented as a function $\boldsymbol{F}_\Theta \colon \mathbb{R}^n \to \mathbb{R}^m$ with trainable parameters Θ:

$$\hat{\boldsymbol{x}} = \boldsymbol{F}_\Theta(\boldsymbol{y})\,. \tag{5.1}$$

where, $\hat{\boldsymbol{x}}$ is the predicted image.

Most of the works in direct reconstruction for sparse-view CT represent $\boldsymbol{F}$ with a neural network with fully-connected layers. These networks require huge memory and large datasets for training. As an alternative to this, we propose the use of fully convolutional encoder-decoder networks that have lesser trainable parameters and are faster to train. The main idea is to enforce data consistency by providing estimates at different resolutions $\hat{\boldsymbol{x}}_r$, $r = 1, \ldots, R$:

$$\hat{\boldsymbol{x}} = \boldsymbol{F}_\Theta(\boldsymbol{y}, (\hat{\boldsymbol{x}}_r)_{r=1}^R) \tag{5.2}$$

where each $\hat{\boldsymbol{x}}_r \in \mathbb{R}^{m_r}$, $m_r < m$, is an approximate solution of

$$\boldsymbol{y} = \boldsymbol{P}\boldsymbol{U}_r\hat{\boldsymbol{x}}_r \tag{5.3}$$

with $\boldsymbol{U}_r \in \mathbb{R}^{m \times m_r}$ being an upsampling operator.

In a typical CED, the encoder learns the representation of the input domain and the decoder learns to map this representation to the corresponding image in the output domain. In the specific case of a CED for medical image reconstruction, the encoder operates in the sinogram space and the decoder in the image space. Based on this hypothesis, we propose to concatenate the estimates at different levels of the decoder part of the network. The function of these concatenations is to help the network learn the structure of the image. The feature maps at different levels of the decoder have different resolutions. Hence, concatenating the estimate $\hat{\boldsymbol{x}}_r$ at different levels requires the estimate to be of the appropriate resolution. The different convolutional layers in the decoder work towards arriving at a clear reconstructed image that is free of artifacts and noise. The estimate $\hat{\boldsymbol{x}}_r$ is obtained with a sparse

sinogram, hence it is artifact-ridden and noisy. Therefore, concatenating the estimate $\hat{\boldsymbol{x}}_r$ at a level closer to the output resolution is counter productive as the network has lesser number of convolutional layers to correct the noise and artifacts. On the other hand the estimate at lower resolutions has lesser structural information compared to the estimates at higher resolution. The selection of $\hat{\boldsymbol{x}}_r$ should ensure a balance between aiding the network to learn the structure of the image and enabling it to correct the artifacts and noise.

Our method, namely LRRCED, was implemented with $R = 2$ and the image estimates $\hat{\boldsymbol{x}}_r$ were obtained by FBP at lower resolution. With the help of a series of experiments, we determined the best possible configuration for concatenating $\hat{\boldsymbol{x}}_r$. In section Section 5.7.4, we present quantitative evaluation of the effect of these concatenations on the reconstructed images.

We investigate LRRCED with two different variations for $\boldsymbol{F}$, LRRCED(D) with Fully Convolutional DenseNets and LRRCED(U) with U-Net, which are discussed in Section 5.2.1 and Section 5.2.1.

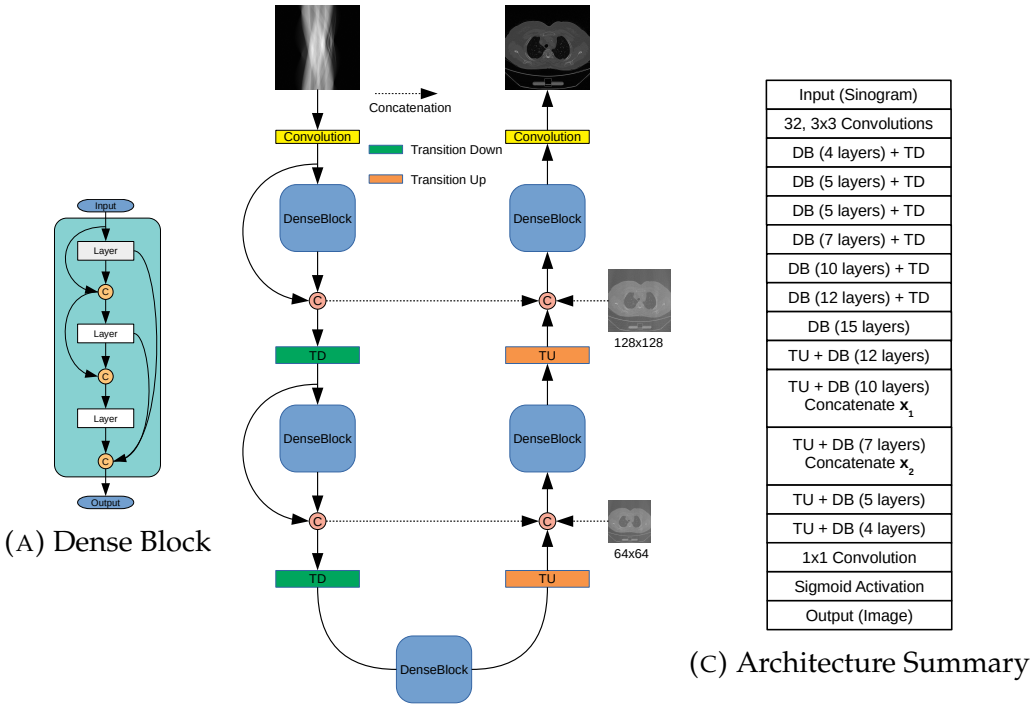
Fully Convolutional Dense Networks

A fully convolutional dense network was used as first variation of LRRCED. Dense networks Huang et al., 2017 are based on the hypothesis that connecting all the layers to each other in a feed forward fashion leads to higher accuracy and easier training of the network. A typical dense block of three layers is depicted in Figure 5.2(a). The extension of dense networks for image segmentation was proposed by Jégou et al., 2017. The three blocks involved in the construction of this network are Dense Block (DB) with l number of layers, Transition Up (TU) and Transition Down (TD). The combination of these three blocks helps in building an encoder-decoder structure suitable for tasks dealing with image-to-image domain transfer. Each layer consists of batch normalization, ReLU activation and 3×3 convolution. TD includes: batch normalization, ReLU, 1×1 Convolution and 2×2 max pooling. Finally, TU includes a 3×3 transposed convolution with stride 2. The important modification to the architecture blocks in our work is the removal of the dropout layers. The fully convolutional dense network with proposed concatenations is represented in Figure **??**. For the sake of representation we included only 5 dense blocks in the figure. The complete architecture details are given in Table **??**.

U-Net

One of the most established architectures for image-to-image translation is U-Net Ronneberger, Fischer, and Brox, 2015, which we used as second variation of LRRCED (called from here on-wards as LRRCED(U)).

A typical U-Net consists of Convolution, Activation (ReLU) and Pooling layers in the encoder and Upsampling, Convolution and Activation in the decoder. We have used U-Net without the dropout, similar to the dense network. The U-Net is represented in Figure 5.3(a).



(A) Dense Block

(B) LRRCED(D)

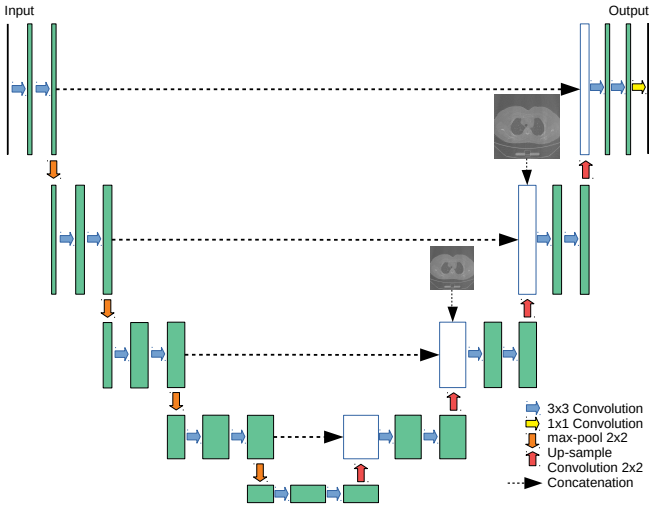
(C) Architecture Summary

FIGURE 5.2: Different components of LRRCED(D): (a) Representation of a dense block with three layers. (b) LRRCED(D): Fully convolutional dense network with x_1 at 64×64 and x_2 at 128×128. (c) Complete architecture summary

Loss Function

The aim of a supervised data-driven image reconstruction task is to predict an image that is as close as possible to the GT image. The appropriate loss function to achieve this is the MAE which is defined as follows:

$$\mathrm{MAE}(\boldsymbol{x}^\star, \hat{\boldsymbol{x}}) = \frac{1}{m} \sum_{j=1}^{m} |x_j^\star - \hat{x}_j| \tag{5.4}$$



(A) LRRCED(U)

Input (Sinogram)
3 layers of 32, 3x3 Convolutions
Max pooling
3 layers of 64, 3x3 Convolutions
Max pooling
3 layers of 128, 3x3 Convolutions
Max pooling
3 layers of 256, 3x3 Convolutions
Max pooling
3 layers of 512, 3x3 Convolutions
256, 3x3 Transposed Convolutions
2 layers of 256, 3x3 Convolutions
128, 3x3 Transposed Convolutions
Concatenate $\mathbf{x}_1$
2 layers of 128, 3x3 Convolutions
64, 3x3 Transposed Convolutions
Concatenate $\mathbf{x}_2$
2 layers of 64, 3x3 Convolutions
32, 3x3 Transposed Convolutions
2 layers of 32, 3x3 Convolutions
1x1 Convolution
Output (Image)

(B) Architecture Summary

FIGURE 5.3: Different components of LRRCED(U): (a) LRRCED(U): U-Net with x_1 at 64×64 and x_2 at 128×128. (b) Complete architecture summary.

where $\boldsymbol{x}^\star = [x_1^\star, \ldots, x_m^\star]^\top \in \mathbb{R}^m$ and $\hat{\boldsymbol{x}} = [\hat{x}_1, \ldots, \hat{x}_m]^\top \in \mathbb{R}^m$ are respectively the true image and predicted image.

In order to improve the resolution of reconstructed images, many deep learning approaches have used the perceptual loss as proposed by Johnson, Alahi, and Fei-Fei, 2016. This loss uses a pre-trained neural network to extract features from the predicted image and the GT. It can be defined as follows:

$$P_k(\boldsymbol{x}^\star, \hat{\boldsymbol{x}}) = |[\text{VGG16}]_k(\boldsymbol{x}^\star) - [\text{VGG16}]_k(\hat{\boldsymbol{x}})|, \quad k = 1, \ldots, 5 \tag{5.5}$$

where $[\text{VGG16}]_k(\boldsymbol{x}^\star)$ and $[\text{VGG16}]_k(\hat{\boldsymbol{x}})$ are the features extracted from block k of the VGG16 neural network Simonyan and Zisserman, 2014 with respectively the GT and the predicted image as inputs. The features extracted from higher layers of the neural network contain generic information (edges, contrast, etc.) while the deeper layers have finer task-specific details. The VGG16 network was pre-trained on Image-Net data Deng et al., 2009 which is far from a medical context. Hence, the higher-level generic features were found to be more relevant for the task of medical image reconstruction. We observed that using extracted features from two different levels, namely Block 1 and Block 3, of the VGG16 network proved to be most effective.

The final loss function that was used for training both the aforementioned networks is defined as follows:

$$\mathcal{L}(\boldsymbol{x}^\star, \hat{\boldsymbol{x}}) = \alpha \text{MAE}(\boldsymbol{x}^\star, \hat{\boldsymbol{x}}) + \beta(P_1(\boldsymbol{x}^\star, \hat{\boldsymbol{x}}) + P_3(\boldsymbol{x}^\star, \hat{\boldsymbol{x}})) \tag{5.6}$$

where P_1 and P_3 are perceptual loss from the extracted features of the two different blocks above-mentioned, α and β are weights which were set to 10 and 0.5 during the training phase.

5.3 Dataset

The data used in this work is from the Large-Scale CT and PET/CT Dataset for Lung Cancer Diagnosis (Lung-PET-CT-Dx) Li et al., 2020b; Clark et al., 2013. Details of the dataset are given in Table **??**. The images in this dataset were reconstructed using FBP on full-angular coverage measurement data. We used the ASTRA toolbox Van Aarle et al., 2016, for data processing to create the projection-image pairs. A fan-beam geometry with a source to detector distance at 1500 mm and source to the center of the rotation at 1000 mm were considered. The number of detectors was set to 700 and the number of

angles was varied to generate different levels of sparsity ($N_a = 60, 90$ and 120). The noise-free projection data were obtained using the Beer-Lambert law (1.16) with an input emission intensity of 10^5. The final projection data were obtained by adding Poisson noise (i.e., (1.12)) to the noise-free projection data. We finally generated the FBP estimates from the noise-added sparse-projections which were used in training the networks as explained previously. Sample images from the dataset are shown in Figure 5.4.

TABLE 5.1: Dataset Description

Dataset Statistics	
Modalities	CT
Number of Participants	355
Number of Studies	436
Number of Series	1295
Number of 2-D Image slices	251,135
CT Matrix size	512

5.4 Training

We implemented the architectures described in the previous section using TensorFlow and Keras Abadi et al., 2016; Chollet, 2015. A subset of the dataset consisting of 22,000 2-D CT images was used in this study. We then split the data into 30,000 images for training and 2,000 images for testing. The sinograms and FBP estimates were generated using the ASTRA toolbox as described above. The sinograms were resized to 512×512 to ensure symmetry with the images for easier training of the network. The FBP estimates $\hat{\boldsymbol{x}}_1$ and $\hat{\boldsymbol{x}}_2$ were resized to the resolutions required for concatenation to the proposed networks. The neural networks were independently trained for each of the sparse-view settings with $N_a = 20, 40, 60, 90$ and 120. The choice of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ were at 64×64 and 128×128 resolutions for LRRCED(D) and 128×128 and 256×256 resolutions for LRRCED(U). The networks were trained for 50 epochs with Adam optimizer with a decay of 10^{-4}.

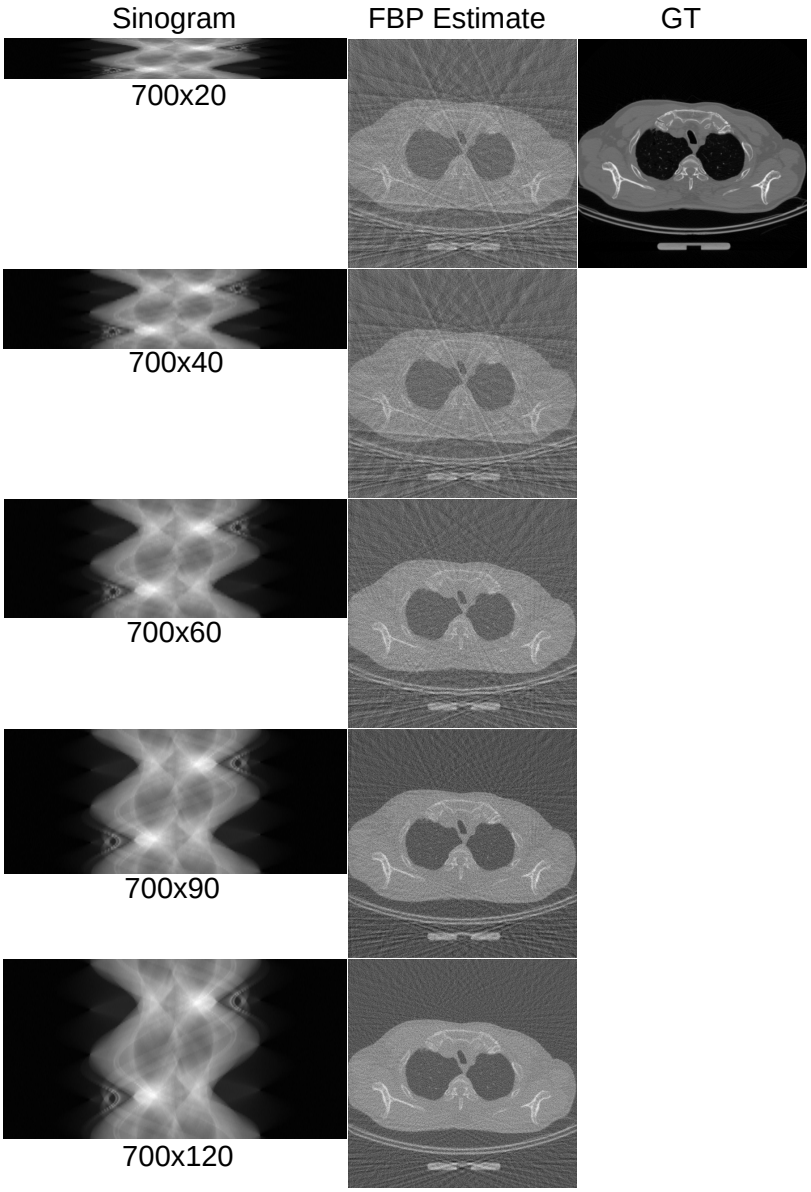


FIGURE 5.4: Samples from the dataset: Sinograms with different sparse-view configurations along with their corresponding FBP estimate.

5.5 Quantitative Analysis:

The metrics used for evaluating the reconstructed images were SSIM and PSNR. They are defined as follows:

$$\text{SSIM}(\boldsymbol{x}^\star, \boldsymbol{x}) = \frac{(2\mu_{\boldsymbol{x}^\star}\mu_{\boldsymbol{x}} + c_1)(2\sigma_{\boldsymbol{x}^\star\boldsymbol{x}} + c_2)}{(\mu_{\boldsymbol{x}^\star}^2 + \mu_{\boldsymbol{x}}^2 + c_1)(\sigma_{\boldsymbol{x}^\star}^2 + \sigma_{\boldsymbol{x}}^2 + c_2)} \tag{5.7}$$

where $\mu_{\boldsymbol{x}^\star}$ and $\mu_{\boldsymbol{x}}$ are the mean of $\boldsymbol{x}^\star$ and $\boldsymbol{x}$ respectively, $\sigma_{\boldsymbol{x}^\star}^2$ and $\sigma_{\boldsymbol{x}}^2$ are the variance of $\boldsymbol{x}^\star$ and $\boldsymbol{x}$, $\sigma_{\boldsymbol{x}^\star\boldsymbol{x}}$ is the covariance between $\boldsymbol{x}^\star$ and $\boldsymbol{x}$, $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$ where $k_1 = 0.01$ and $k_2 = 0.03$ by default,

$$\text{PSNR} = 20\log_{10}\left(\frac{L-1}{\text{RMSE}}\right) \tag{5.8}$$

where L is the maximum intensity in the image and RMSE is given by

$$\text{RMSE}(\boldsymbol{x}^\star, \hat{\boldsymbol{x}}) = \sqrt{\frac{1}{m}\sum_{j=1}^{m}(x_j^\star - \hat{x}_j)^2}\,. \tag{5.9}$$

5.6 Comparative Analysis

The LRRCED method was compared with a post-processing deep learning-based approach, namely FBP-ConvNet Jin et al., 2017, and a penalized weighted least-squares (PWLS)-TV solver for the model-based iterative CT reconstruction Tang, Nett, and Chen, 2009. We trained FBP-ConvNet on a set of 30,000 noisy, artifact-ridden FBP image and GT pairs. This network was trained for 50 epochs.

5.7 Results

5.7.1 Experimental Results

Fig. 5.5 shows the images reconstructed with LRRCED(D) for various degrees of sparsity in the projections. Images from various parts of the patient volume are displayed at different HUT windows for clearer evaluation of the proposed approach. We observe the improvement in the reconstructed images with the decrease in sparsity in the views. The images reconstructed with $N_\text{a} = 120$ appear closest to the GT. The soft tissue regions in the images reconstructed with <60 views show artifacts which are not present with

the use of more projections. Similarly in Fig. 5.6, we show the images reconstructed with LRRCED(U).

In Fig. 5.7 and Fig. 5.8 we present a comparison of reconstructed images using different algorithms with 60 and 90 views respectively. The top row consists of the GT and the reconstructed image by proposed LRRCED(D) approach. The second row consists of images with LRRCED(U) and the FBP-ConvNet. Finally in the last row are the images reconstructed with PWLS-TV iterative method and FBP. The region highlighted in yellow is zoomed and displayed alongside the corresponding image. These methods are quantitatively compared in Table 5.2 and Table 5.3. We observe that the deep learning methods perform better than the iterative and analytical methods. The images reconstructed with U-Net based methods namely LRRCED(U) and FBP-ConvNet, have very similar characteristics: The contrast is higher and they perform better quantitatively. However, images reconstructed with DenseNet by comparison show less noise and streaking artifacts. These visual observations can be more clearly seen in the zoomed images shown in Fig. 5.7. This is further reiterated in the intensity plot profiles shown in Fig. 5.9 and Fig. 5.10, where the LRR-CED(D) results are closer to the GT. In accordance with the metrics tabulated in Table 5.2 and Table 5.3, we find that the plots of deep learning-based methods are very close to that of the GT. Even though the proposed approach with typical CEDs performs a task which is more complex than denoising, the metrics indicate that the quality has not deteriorated compared to a standard post-processing approach.

TABLE 5.2: Quantitative comparison of various reconstruction algorithms with SSIM and PSNR for projections with 60 views

Metric	FBP	PWLS-TV	FBP ConvNet	LRRCED (D)	LRRCED (U)
SSIM	0.16	0.66	0.90	0.89	0.90
PSNR	11.57	28.23	31.58	30.04	30.20

TABLE 5.3: Quantitative comparison of various reconstruction algorithms with SSIM and PSNR for projections with 90 views

Metric	FBP	PWLS-TV	FBP ConvNet	LRRCED (D)	LRRCED (U)
SSIM	0.19	0.72	0.93	0.91	0.92
PSNR	13.57	30.21	35.27	32.70	32.86

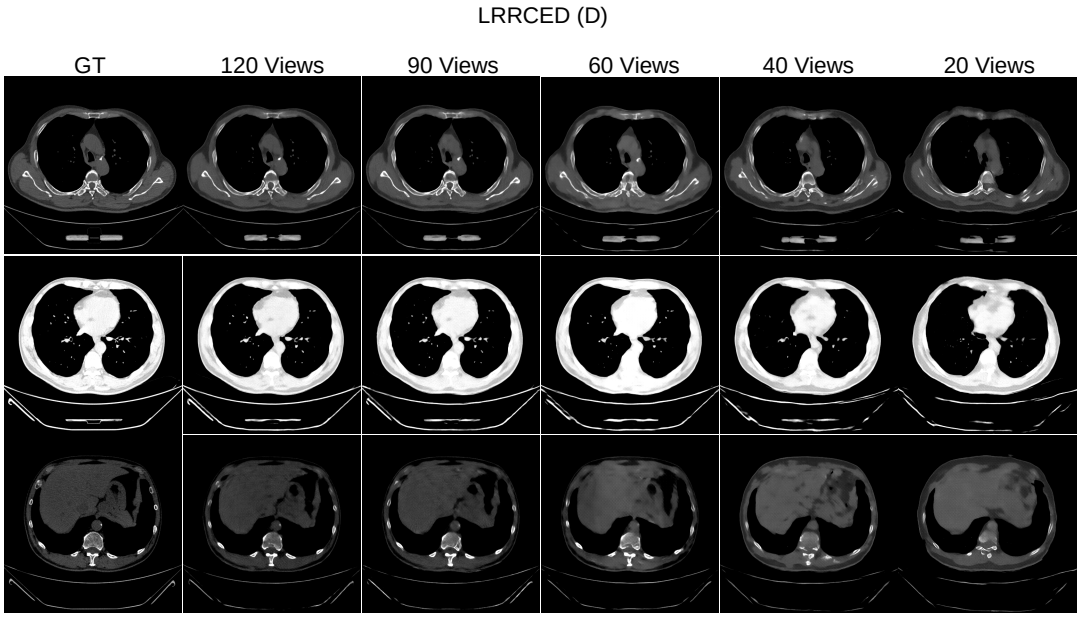


FIGURE 5.5: Images reconstructed with LRR-CED(D) approach with different sparse-view configurations, i.e., projections with $N_a = 120, 90, 60, 40$ and 20. For better visual inspection images in first row are displayed in -40 ± 600 HUT window, the second row in -340 ± 400 HUT and the third in -150 ± 400 HUT.

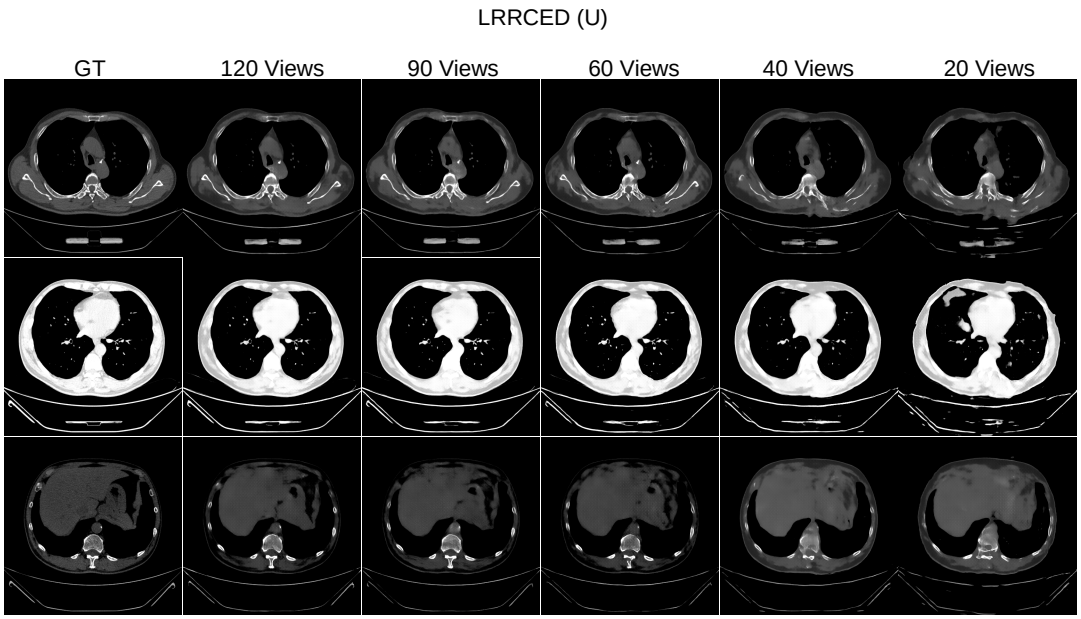


FIGURE 5.6: Images reconstructed with LRR-CED(U) approach with different Sparse-View configurations, i.e., projections with $N_a = 120, 90, 60, 40$ and 20. Images in first row are displayed in -40 ± 600 HUT window, the second row in -340 ± 400 HUT and the third in -150 ± 400 HUT.

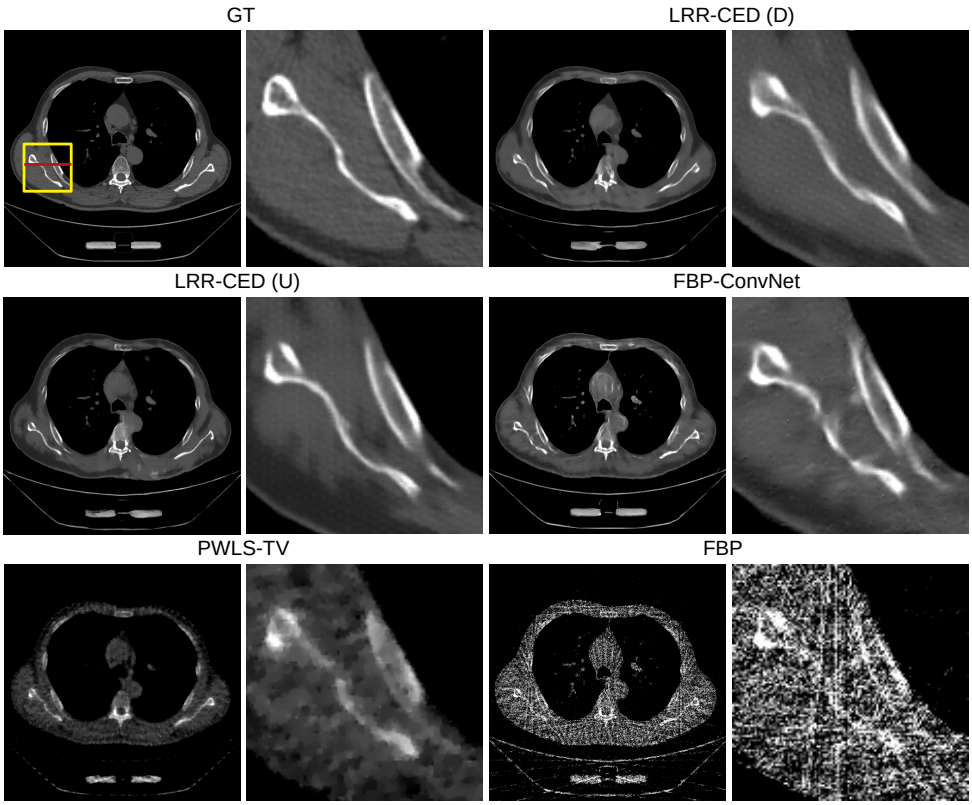


FIGURE 5.7: Comparative analysis for 60 views: From the top left corner, we have GT image, reconstructions with LR-RCED(D) . In the second row reconstructed images with LR-RCED(U) and FBP-ConvNet. Finally images reconstructed with PWLS-TV and FBP.

5.7.2 Experiments with real data

The proposed networks were initialized with the weights from the previous study and were then trained on the real data. The real data used in this study was part of the Low Dose CT grand challenge McCollough, 2016. The data constituted of 10 patients, acquired with flying spot technique and a helical scan. It was a subset of the larger Mayo CT clinic database Moen et al., 2021. The data from nine patients constituting of 3,994 2-D slices was used for training and the trained network was tested on another patient data. The three-dimensional (3-D) sinograms obtained from the helical scan were converted into 2-D sinograms through the single slice re-binning method employed in Kim, El Fakhri, and Li, 2017. We further resampled the sinograms reducing the number of views to 64. The number of detector panels was 734. The FBP estimates were generated from these sparse-view sinograms and resized for training the LRRCED.

We present the results for four different slices across the patient volume and their quantitative evaluation in Figure 5.11 and Table 5.4, respectively.

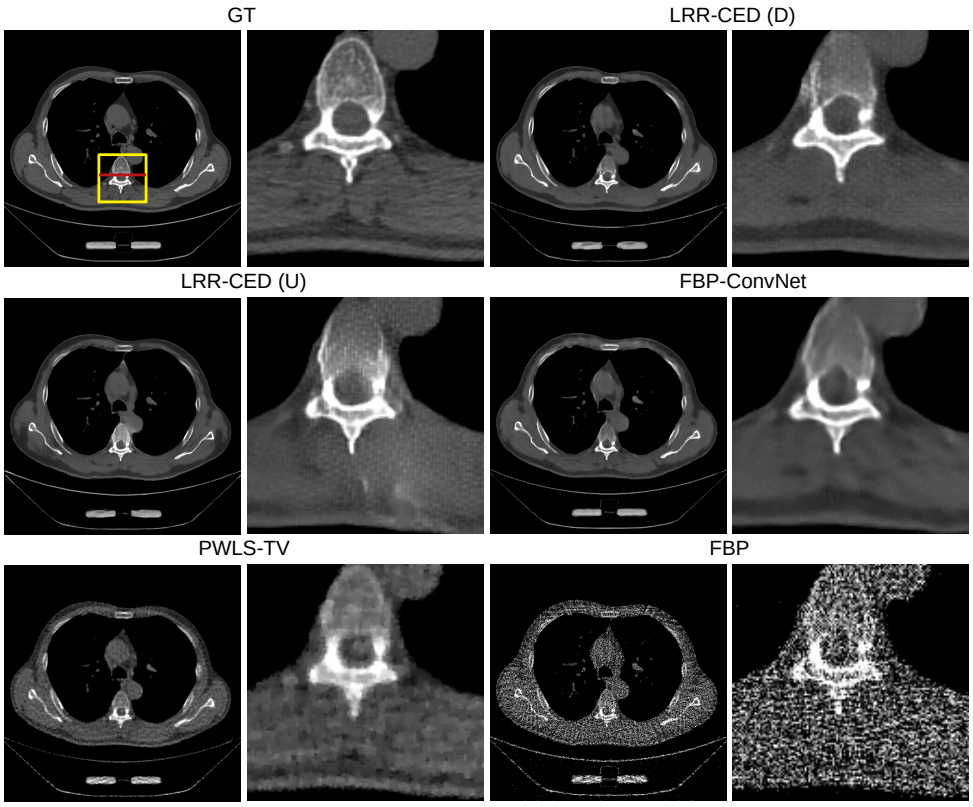
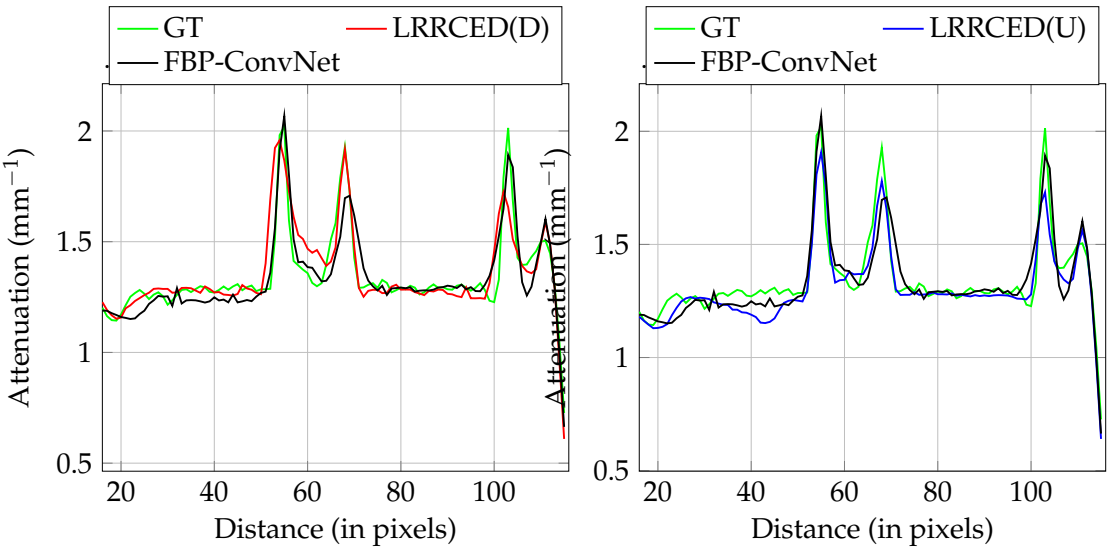


FIGURE 5.8: Comparative analysis for 90 views: From the top left corner, we have GT image, reconstructions with LR-RCED(D) . In the second row reconstructed images with LR-RCED(U) and FBP-ConvNet. Finally images reconstructed with PWLS-TV and FBP.

We observe that the reconstructed images with the proposed networks have similar characteristics as the ones from the simulation study. The transfer learning strategy ensures that the quality of the reconstructed images is maintained even with very limited training data.

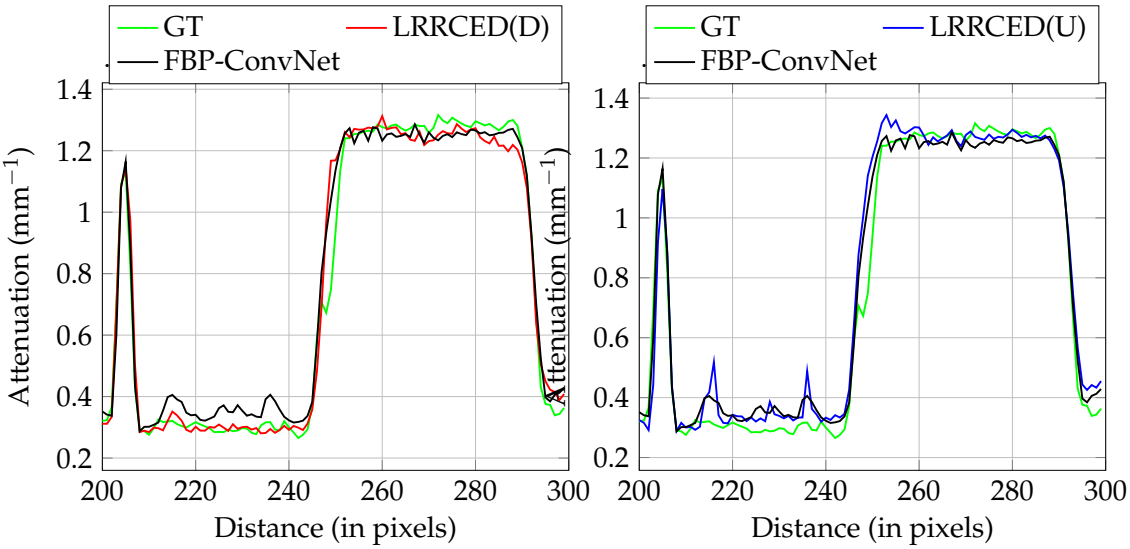
TABLE 5.4: Quantitative comparison of images reconstructed with the proposed algorithms w.r.t. GT across different slices in the patient volume from the real dataset displayed in Fig. 5.11

Image	Metric	LRRCED(D)	LRRCED(U)
a	SSIM	0.89	0.92
	PSNR	35.70	36.64
b	SSIM	0.88	0.92
	PSNR	35.19	36.13
c	SSIM	0.94	0.92
	PSNR	40.86	42.04
d	SSIM	0.84	0.91
	PSNR	33.37	34.59



(A) 60 views 1 (B) 60 views 2

FIGURE 5.9: Intensity plot profile for the region marked in red from Fig. 5.7 comparing LRRCED(D) and FBP-ConvNet to the GT in (a) and LRRCED(U) and FBP-ConvNet in (b)



(A) 90 views 1 (B) 90 views 2

FIGURE 5.10: Intensity plot profile for the region marked in red from Fig. 5.8 comparing LRRCED(D) and FBP-ConvNet to the GT in (a) and LRRCED(U) and FBP-ConvNet in (b)

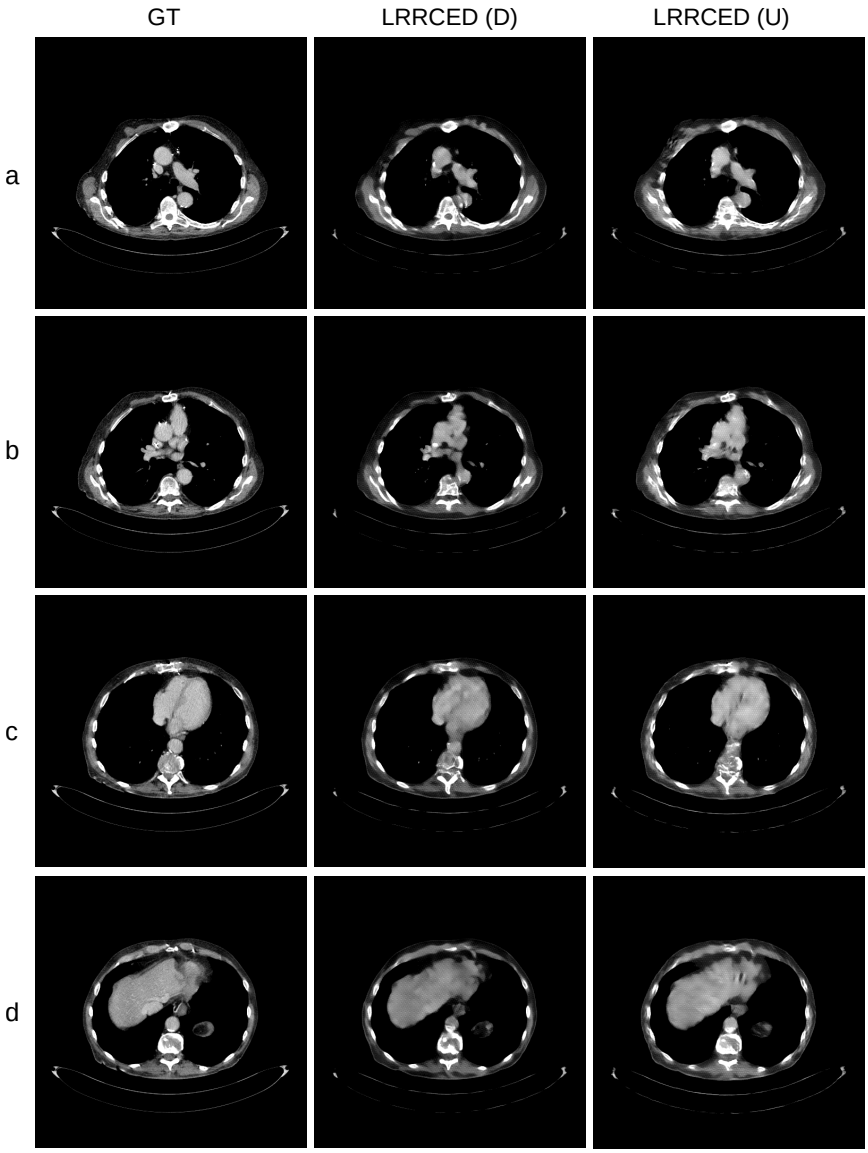


FIGURE 5.11: Real data study: Images reconstructed with the proposed approaches across 4 different slices displayed in the window 40 ± 200 HUT.

5.7.3 Stability Study

One of the major challenges to data-driven neural network approaches is the ability to generalize over different types of test data. The extent to which a neural network is stable when presented with data different from the training data is the focus of this study. This topic has been extensively evaluated in the article by Antun et al., 2020. The authors analyzed the impact of tiny perturbations and small structural changes in sampling and image domain on the reconstructed images. They also observed the way in which a change in sampling (sparsity in CT for example) could influence performance. In our work centered around sparse-view CT image reconstruction, we performed

a series of experiments with different levels of sparsity in the testing data. The proposed network LRRCED(D) was trained separately on each of the sparsity configurations, ($N_a = 20, 40, 60, 90$ and 120). It was then tested using the sinograms and the corresponding FBP estimates for all of the possible values of N_a considered.

The results are displayed in Fig 5.12. The top row corresponds to network trained with 20-view data, the second with 40-view data and so on. The trend is towards an improvement in overall image quality with reduced sparsity in the sinograms. On one hand, we observe that in the scenarios where the testing data has more sparsity than the training data, the artifacts in the reconstructed images are more clearly visible. This is clearly seen in the last two rows in Figure 5.12, where the network was trained on 90 views and 120 views data and the images reconstructed with lower N_a are ridden with artifacts. On the other hand, the image quality especially in the soft tissue regions is higher when the network is trained and tested on data with more views. The proposed network maintains stability in the reconstructed images with the increase in the sampling in the testing data. However, when the testing data has fewer views than the training data, artifacts are present in the reconstructed images.

5.7.4 Hyperparameter optimization

Finding the optimal hyperparameters is an important aspect of training neural networks. The common hyperparameters in a typical CNN are number of filters, number of layers, etc. These interdependent hyperparameters determine the rate of convergence and require task-specific experimentation to arrive at the best possible configuration. The unique hyperparameters in our proposed approach are the resolutions of concatenated FBP estimates. The number of training examples is another important component that varies depending on the task and the trainable parameters of the neural network selected for the task. In this section we discuss our experiments that determined the selection of these two important hyperparameters.

Concatenation Resolution Selection

To select the best possible configuration for concatenation in the proposed approach, we trained the networks with a fixed set of hyper-parameters and different combinations of concatenations. We discuss the results with LRRCED(D) in this regard. The number of training samples were set to 10,000

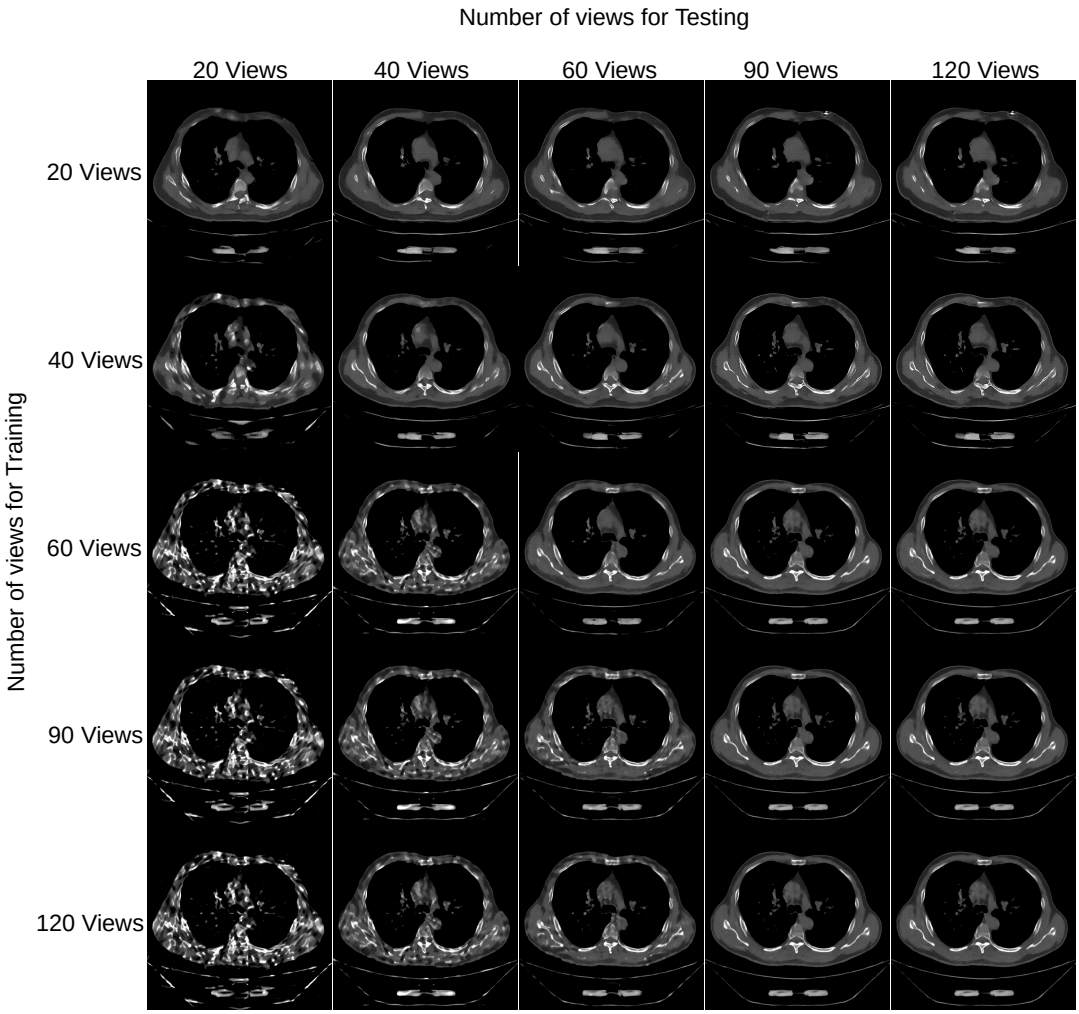


FIGURE 5.12: Stability study: Each row corresponds to the network trained on specific value of N_a, and tested with all the possible values of N_a.

for all the experiments. The training data were projections with 90 views, corresponding FBP reconstructed images and the GT. The training was done for 25 epochs. Each of the concatenation setting was evaluated on 5 test patients. The average SSIM for each patient was plotted for each of the experiment setting. In Fig 5.13 we have the average SSIM vs Patient plot for single concatenation at a specific resolution. Similarly Figure 5.14 consists of plots for double concatenation at two different resolutions. The double concatenation at $64 \times 64, 128 \times 128$ overall leads to the best metrics, thus becoming our choice for the experiments in this work. These results are tabulated in Table 5.5.

Training Examples Analysis

One of the biggest challenges in any data driven algorithm is the selection of training examples required for the experiments. It is important to analyze this hyper-parameter as it serves as an important factor for the network to be reproducible and scalable. We varied the number of training examples for the best concatenation setting from the previous section and the 90-view scenario. The evaluation was similar to the previous experiment with the average SSIM for 5 patients. The results from these experiments are tabulated in Table 5.6. As seen in Figure 5.15, the performance of the network improves along with the increase in the number of training examples. There is however a marginal difference in the performance of the network when trained with 20,000 or 30,000 training examples, hence making us choose 20,000 training examples as the optimum number for this hyper-parameter. The average SSIM values across the test patients tend to get similar as the number of training examples increases.

5.7.5 Ablation Study

We performed an ablation study to understand the impact of the proposed concatenations on the neural network performance. DenseNet described earlier was trained for 50 epochs on 20,000 data samples in three different scenariosas shown in Figure 5.16, two of which used either a sinogram consisting of randomly distributed Gaussian noise and no low-resolution concatenations: (i) true sinogram and the reconstructed image only (no low-resolution concatenations), (ii) Gaussian noise sinogram, low-resolution concatenations and the reconstructed images, and (iii) true sinogram, low-resolution concatenations and the reconstructed images.

The image predictions by the three different neural networks are shown in Figure 5.17. DenseNet without the low-resolution concatenations does produce images with some structural information, but the other two configurations generate images of much better quality. We observe that the concatenations indeed help the network learn the structure of the image, while the sinograms contribute in artifact and noise removal. This is reflected upon closer inspection of the third and fourth images in Figure 5.17. The images predicted with LRRCED(D) trained using the randomly distributed Gaussian noise sinogram instead of the true sinogram have artifacts and noise which is also seen quantitatively in Table 5.7. The best metrics and image quality

are demonstrated by the neural network trained on the combination of sinograms and low-resolution estimates labeled as LRRCED(D) in Figure 5.17.

TABLE 5.5: Average SSIM for different configurations of concatenations

Concatenated FBP Resolution	Average SSIM				
	P1	P2	P3	P4	P5
(32×32)	0.82	0.86	0.88	0.86	0.80
(64×64)	0.85	0.88	0.90	0.88	0.82
(128×128)	0.85	0.87	0.90	0.89	0.81
(256×256)	0.58	0.88	0.85	0.88	0.79
(512×512)	0.66	0.78	0.82	0.75	0.73
$(32 \times 32, 64 \times 64)$	0.83	0.77	0.80	0.80	0.68
$(\mathbf{64 \times 64, 128 \times 128})$	**0.85**	**0.88**	**0.91**	**0.89**	**0.83**
$(128 \times 128, 256 \times 256)$	0.67	0.78	0.83	0.84	0.70

TABLE 5.6: Average SSIM for different number of training examples

Number of Training examples	Average SSIM				
	P1	P2	P3	P4	P5
$1,000$	0.82	0.79	0.86	0.85	0.72
$5,000$	0.84	0.77	0.86	0.84	0.69
$10,000$	0.85	0.88	0.91	0.89	0.83
$\mathbf{20,000}$	**0.89**	**0.90**	**0.91**	**0.90**	**0.82**
$30,000$	0.89	0.89	0.90	0.90	0.82

TABLE 5.7: Ablation Study: Quantitative comparison of different configurations of the DenseNet

Sl.No.	True sinograms	Concatenations	Gaussian noise sinograms	SSIM	PSNR
(i)	✓	✗	✗	0.29	12.05
(ii)	✗	✓	✓	0.70	28.89
(iii)	✓	✓	✗	**0.88**	**32.53**

5.8 Discussion

The use of deep learning architectures in the framework of medical image reconstruction is propelled by potentially faster reconstruction without compromising on the quality of the images. To this end, hybrid image reconstruction involving unrolled iterative algorithms with embedded deep learning

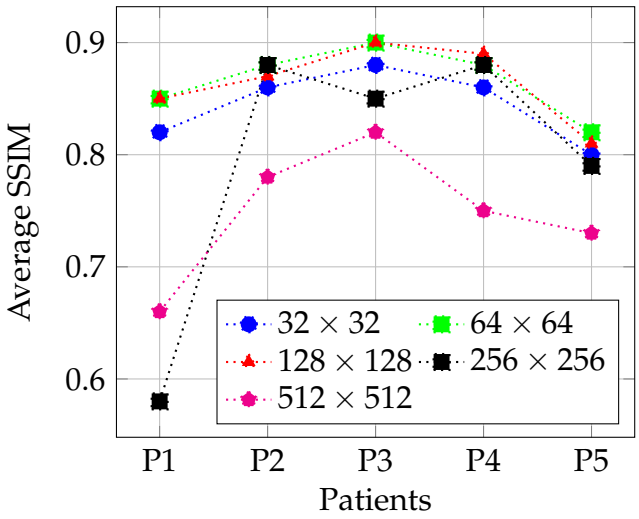


FIGURE 5.13: Comparison of single concatenations for the particular case of 90 views evaluated with SSIM on 5 different patients from the dataset. The best metrics are found with concatenation at 128×128.

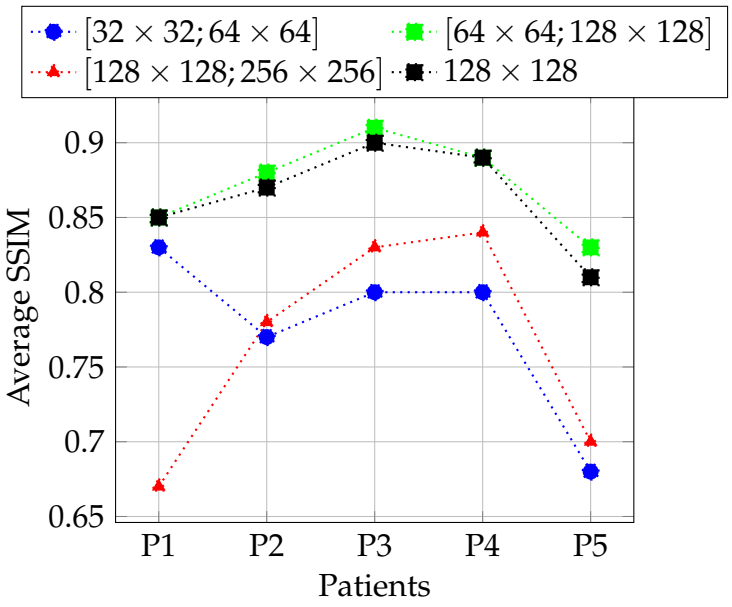


FIGURE 5.14: Comparison of double concatenations for the particular case of 90 views evaluated with SSIM on 5 different patients from the dataset. The best metrics are found with concatenations at 64×64 and 128×128 resolutions.

architectures do not significantly reduce the reconstruction time. Hence, the use of deep learning architectures for either improving images from a fast analytic algorithm or direct reconstruction becomes more relevant for their incorporation into the image reconstruction pipeline. One significant problem for direct image reconstruction is the requirement of large and complex networks to learn the mapping from sinograms to images without the help

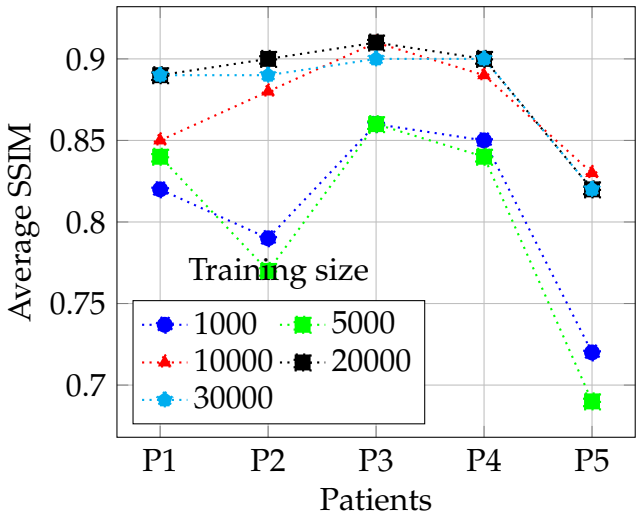


FIGURE 5.15: Comparison of Average SSIM for 5 different Patient data for 90 views with varying number of training samples. The configuration of the network is the one with best performance from the analysis in Figure 5.13. (concatenations at 64×64 and 128×128).

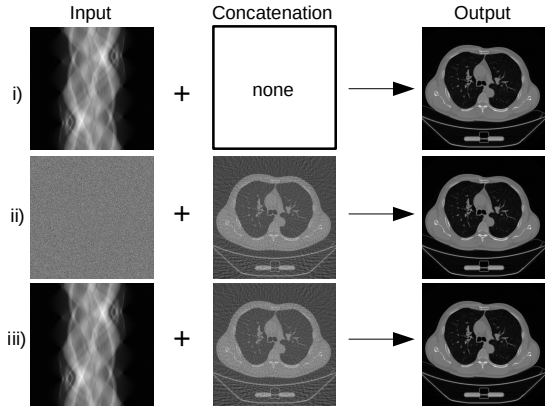


FIGURE 5.16: Schematic representation of configurations used in the ablation study: (i) true sinogram and the reconstructed image only (no low-resolution concatenations); (ii) randomly distributed Gaussian noise sinogram, low-resolution concatenations and the reconstructed images; (iii) true sinogram, low-resolution concatenations and the reconstructed images.

of any reconstruction estimate. The networks used for post-processing on the other hand are simpler and relatively easy to train. In this work we attempted to use these post-processing networks for the direct image reconstruction task along with low-resolution scout images from direct analytical method. We show that concatenating FBP estimates at lower resolutions is sufficient to allow the network to learn the mapping from sinogram to image space. Through the use of two different networks with the concatenation approach we demonstrate that this idea can be applied to CEDs in general.

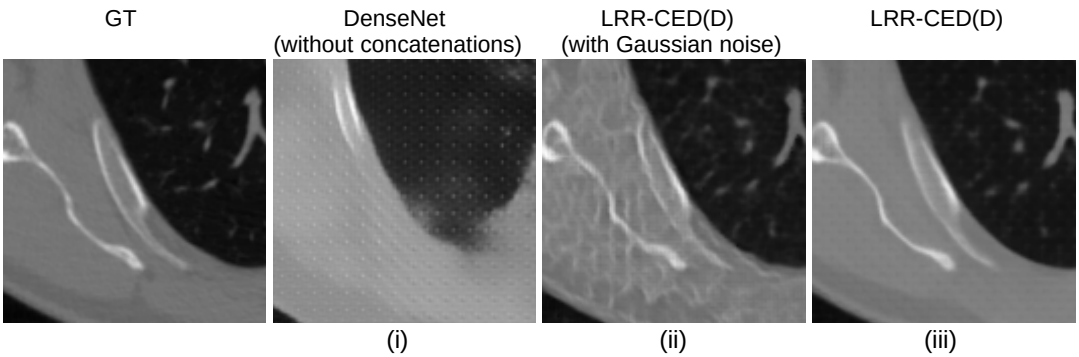


FIGURE 5.17: Ablation study: Predictions from different configurations of the network.

In the sparse-view CT scenario artifact removal along with denoising increases the challenges of getting a clean well-resolved image. We observed that the use of traditional loss functions (L1 or L2) resulted in blurry images. To tackle this and to improve the sharpness of the images we used perceptual loss along with the standard L1 loss. The reconstructed images with our proposed LRRCED(D) and LRRCED(U) have higher SSIM and PSNR than images reconstructed with a traditional iterative algorithm and a standard post-processing deep learning method FBP-ConvNet. The similarity in the images from the deep learning methods stems from the fact that the choice of networks used in our proposed work was inspired from post-processing CEDs. The contribution in this work is the use of these networks to learn the mapping from sparse sinograms to images with the same amount of training examples, which is possible only with the proposed addition of the concatenations. Through the ablation study from Section 5.7.5, we reiterate the contribution of both the sinogram and the low-resolution concatenations for image reconstruction. The CED without the concatenations could learn the mapping but it would need much higher number of training examples for image quality comparable to other methods. The proposed method was compared to a U-Net based denoising method (FBP-ConvNet), which has one of the best quantitative metrics in image reconstruction as established by the recent quantitative comparison study carried out by Leuschner et al., 2021. As it was shown in this study, complex unrolled methods do only marginally better than the U-Net, hence making it one of the most frequently used benchmarks for comparison purposes.

We are currently exploring the possibility of using image estimates from earlier iterations of standard iterative algorithms while ensuring that the trade-off between time and image quality is not compromised. The use of other alternative architectures is also being explored to arrive at reconstructed

images which perform significantly better than existing post-processing approaches. Finally, we are working on experiments with low-dose CT and other tomographic reconstruction modalities to establish the adaptability of the proposed approach.

5.9 Conclusion

In this work we studied the use of fully convolutional encoder-decoder networks in direct sparse-CT image reconstruction. We introduced a new approach that uses lower dimension FBP estimates as concatenations to help the network learn the mapping from sinogram to image space. In the context of image reconstruction, we inject the information from the inverse of a CT physical system (FBP estimate) as a feature map in the decoder. We presented two variations of the proposed approach namely LRRCED(D) using fully convolutional dense networks and LRRCED(U) using U-Net. The proposed neural networks reconstruct images that are either better or are on par with traditional reconstruction algorithms and post-processing deep learning based approach (FBP-ConvNet). A single pass of a sparse sinogram through the network results in reconstructed images without the artifacts and noise which are severely present in the concatenated FBP estimates. Finally, this idea of using task specific concatenations that enable one to have control over what the network learns, can be extended to various other problems in medical imaging.

Appendix A

Frequently Asked Questions

A.1 How do I change the colors of links?

The color of links can be changed to your liking using:

`\hypersetup{urlcolor=red}`, or

`\hypersetup{citecolor=green}`, or

`\hypersetup{allcolor=blue}`.

If you want to completely hide the links, you can use:

`\hypersetup{allcolors=.}`, or even better:

`\hypersetup{hidelinks}`.

If you want to have obvious links in the PDF but not the printed text, use:

`\hypersetup{colorlinks=false}`.

Bibliography

[Aba+16] Martín Abadi et al. "Tensorflow: A system for large-scale machine learning". In: *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*. 2016, pp. 265–283.

[ACB17] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein generative adversarial networks". In: *International conference on machine learning*. PMLR. 2017, pp. 214–223.

[Amy+19] A Amyar et al. "3-D RPET-NET: development of a 3-D PET imaging convolutional neural network for radiomics analysis and outcome prediction". In: *IEEE Transactions on Radiation and Plasma Medical Sciences* 3.2 (2019), pp. 225–231.

[Ant+20] Vegard Antun et al. "On instabilities of deep learning in image reconstruction and the potential costs of AI". In: *Proceedings of the National Academy of Sciences* 117.48 (2020), pp. 30088–30095.

[AÖ18] Jonas Adler and Ozan Öktem. "Learned primal-dual reconstruction". In: *IEEE Transactions on Medical Imaging* 37.6 (2018), pp. 1322–1332.

[Che+17] Hu Chen et al. "Low-dose CT denoising with convolutional neural network". In: *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*. IEEE. 2017, pp. 143–146.

[Cho+15] François Chollet et al. *Keras*. https://github.com/fchollet/keras. 2015.

[Cla+13] Kenneth Clark et al. "The Cancer Imaging Archive (TCIA): maintaining and operating a public information repository". In: *Journal of Digital Imaging* 26.6 (2013), pp. 1045–1057.

[Cui+18] Sunan Cui et al. "Artificial Neural Network With Composite Architectures for Prediction of Local Control in Radiotherapy". In: *IEEE Transactions on Radiation and Plasma Medical Sciences* 3.2 (2018), pp. 242–249.

[Den+09] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[Dol+18] Jose Dolz et al. "HyperDense-Net: a hyper-densely connected CNN for multi-modal image segmentation". In: *IEEE Transactions on Radiation and Plasma Medical Sciences* 38.5 (2018), pp. 1116–1126.

[DP95] A. R. De Pierro. "A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography". In: *IEEE Transactions on Medical Imaging* 14.1 (1995), pp. 132–137.

[DV16] Vincent Dumoulin and Francesco Visin. "A guide to convolution arithmetic for deep learning". In: *arXiv preprint arXiv:1603.07285* (2016).

[EF02] I. A. Elbakri and J. A. Fessler. "Statistical image reconstruction for polyenergetic X-ray computed tomography". In: *IEEE Transactions on Medical Imaging* 21.2 (2002), pp. 89–99.

[FDM19] Lin Fu and Bruno De Man. "A hierarchical approach to deep learning and its application to tomographic reconstruction". In: *15th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*. Vol. 11072. International Society for Optics and Photonics. 2019, p. 1107202.

[FSF00] J. A. Fessler, M. Sonka, and J. M. Fitzpatrick. "Statistical image reconstruction methods for transmission tomography". In: *Handbook of medical imaging* 2 (2000), pp. 1–70.

[Gon+18] Kuang Gong et al. "PET image denoising using a deep neural network through fine tuning". In: *IEEE Transactions on Radiation and Plasma Medical Sciences* 3.2 (2018), pp. 153–161.

[Gon+19] Kuang Gong et al. "Iterative PET image reconstruction using convolutional neural network representation". In: *IEEE Transactions on Medical Imaging* 38.3 (2019), pp. 675–685.

[Guo+16] Yanming Guo et al. "Deep learning for visual understanding: A review". In: *Neurocomputing* 187 (2016), pp. 27–48.

[Guo+19] Zhe Guo et al. "Deep learning-based image segmentation on multimodal medical imaging". In: *IEEE Transactions on Radiation and Plasma Medical Sciences* 3.2 (2019), pp. 162–169.

[Gup+18] Harshit Gupta et al. "CNN-based projected gradient descent for consistent CT image reconstruction". In: *IEEE Transactions on Medical Imaging* 37.6 (2018), pp. 1440–1453.

[GVGS16] Hayit Greenspan, Bram Van Ginneken, and Ronald M Summers. "Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique". In: *IEEE Transactions on Medical Imaging* 35.5 (2016), pp. 1153–1159.

[Hae+18] Ida Haeggstroem et al. "DeepRec: A deep encoder-decoder network for directly solving the PET reconstruction inverse problem". In: *arXiv preprint arXiv:1804.07851* (2018).

[Hat+18] Mathieu Hatt et al. "The first MICCAI challenge on PET tumor segmentation". In: *Medical image analysis* 44 (2018), pp. 177–195.

[HSW90] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks". In: *Neural networks* 3.5 (1990), pp. 551–560.

[Hua+17] Gao Huang et al. "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[Iso+17] Phillip Isola et al. "Image-to-image translation with conditional adversarial networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1125–1134.

[JAFF16] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. "Perceptual losses for real-time style transfer and super-resolution". In: *European conference on computer vision*. Springer. 2016, pp. 694–711.

[Jin+17] Kyong Hwan Jin et al. "Deep convolutional neural network for inverse problems in imaging". In: *IEEE Transactions on Image Processing* 26.9 (2017), pp. 4509–4522.

[Jég+17] Simon Jégou et al. "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017, pp. 11–19.

[Kad+18] Venkata S Kadimesetty et al. "Convolutional neural network-based robust denoising of low-dose computed tomography perfusion maps". In: *IEEE Transactions on Radiation and Plasma Medical Sciences* 3.2 (2018), pp. 137–152.

[Kan+20] VSS Kandarpa et al. "DUG-RECON: A Framework for Direct Image Reconstruction Using Convolutional Generative Networks". In: *IEEE Transactions on Radiation and Plasma Medical Sciences* 5.1 (2020), pp. 44–53.

[KB14] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[KEFL17] Kyungsang Kim, Georges El Fakhri, and Quanzheng Li. "Low-dose CT reconstruction using spatially encoded nonlocal penalty". In: *Medical physics* 44.10 (2017), e376–e390.

[Kim+18] Kyungsang Kim et al. "Penalized PET reconstruction using deep learning prior and local linear fitting". In: *IEEE Transactions on Medical Imaging* 37.6 (2018), pp. 1478–1487.

[Kos+15] Lale Kostakoglu et al. "A phase II study of 3-deoxy-3-18F-fluorothymidine PET in the assessment of early response of breast cancer to neoadjuvant chemotherapy: results from ACRIN 6688". In: *Journal of Nuclear Medicine* 56.11 (2015), pp. 1681–1689.

[KRF14] Donghwan Kim, Sathish Ramani, and Jeffrey A Fessler. "Combining ordered subsets and momentum for accelerated X-ray CT image reconstruction". In: *IEEE Transactions on Medical Imaging* 34.1 (2014), pp. 167–178.

[LB+95] Yann LeCun, Yoshua Bengio, et al. "Convolutional networks for images, speech, and time series". In: *The handbook of brain theory and neural networks* 3361.10 (1995), p. 1995.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[Led+17] Christian Ledig 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*. 2017, pp. 4681–4690.

[Lee+18] Hoyeon Lee et al. "Deep-neural-network-based sinogram synthesis for sparse-view CT image reconstruction". In: *IEEE Transactions on Radiation and Plasma Medical Sciences* 3.2 (2018), pp. 109–119.

[Leu+21] Johannes Leuschner et al. "Quantitative Comparison of Deep Learning-Based Image Reconstruction Methods for Low-Dose and Sparse-Angle CT Applications". In: *Journal of Imaging* 7.3 (2021), p. 44.

[Li+19] Yinsheng Li et al. "Learning to reconstruct computed tomography images directly from sinogram data under a variety of data acquisition conditions". In: *IEEE Transactions on Medical Imaging* 38.10 (2019), pp. 2469–2481.

[Li+20a] Meng Li et al. "SACNN: Self-Attention Convolutional Neural Network for Low-Dose CT Denoising with Self-supervised Perceptual Loss Network". In: *IEEE Transactions on Medical Imaging* (2020).

[Li+20b] P. Li et al. *A Large-Scale CT and PET/CT Dataset for Lung Cancer Diagnosis*. data retrieved from The Cancer Imaging Archive., https://doi.org/10.7937/TCIA.2020.NNC2-0461. 2020.

[Lim+17] Bee Lim et al. "Enhanced deep residual networks for single image super-resolution". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017, pp. 136–144.

[Lit+17] Geert Litjens et al. "A survey on deep learning in medical image analysis". In: *Medical image analysis* 42 (2017), pp. 60–88.

[Liu+13] Y. Liu et al. "Total variation-Stokes strategy for sparse-view X-ray CT image reconstruction". In: *IEEE Transactions on Medical Imaging* 33.3 (2013), pp. 749–763.

[Mai+18] Joscha Maier et al. "Deep scatter estimation (DSE): Accurate real-time scatter estimation for X-ray CT using a deep convolutional neural network". In: *Journal of Nondestructive Evaluation* 37.3 (2018), p. 57.

[McC16] C McCollough. "TU-FG-207A-04: Overview of the Low Dose CT Grand Challenge". In: *Medical physics* 43.6Part35 (2016), pp. 3759–3760.

[Moe+21] Taylor R Moen et al. "Low-dose CT image and projection dataset". In: *Medical physics* 48.2 (2021), pp. 902–911.

[Nuy+98] John Nuyts et al. "Iterative reconstruction for helical CT: a simulation study". In: *Physics in Medicine & Biology* 43.4 (1998), p. 729.

[Per+18] Dimitris Perdios et al. “Deep convolutional neural network for ultrasound image enhancement”. In: *2018 IEEE International Ultrasonics Symposium (IUS)*. IEEE. 2018, pp. 1–4.

[Rea+20] Andrew J Reader et al. “Deep learning for PET image reconstruction”. In: *IEEE Transactions on Radiation and Plasma Medical Sciences* 5.1 (2020), pp. 1–25.

[RFB15] 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*. Springer. 2015, pp. 234–241.

[RHW86] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. “Learning representations by back-propagating errors”. In: *nature* 323.6088 (1986), pp. 533–536.

[SD19] Ashish Sinha and Jose Dolz. “Multi-scale guided attention for medical image segmentation”. In: *arXiv preprint arXiv:1906.02849* (2019).

[SV82] L. A. Shepp and Y. Vardi. “Maximum Likelihood Reconstruction for Emission Tomography”. In: *IEEE Transactions on Medical Imaging* 1.2 (1982), pp. 113–122.

[SZ14] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[Tha+18] Franz Thaler et al. “Sparse-view CT reconstruction using wasserstein GANs”. In: *International workshop on machine learning for medical image reconstruction*. Springer. 2018, pp. 75–82.

[TNC09] Jie Tang, Brian E Nett, and Guang-Hong Chen. “Performance comparison between total variation (TV)-based compressed sensing and statistical iterative reconstruction algorithms”. In: *Physics in Medicine & Biology* 54.19 (2009), p. 5781.

[VA+16] Wim Van Aarle et al. “Fast and flexible X-ray tomography using the ASTRA toolbox”. In: *Optics express* 24.22 (2016), pp. 25129–25147.

[Vou+18] Athanasios Voulodimos et al. “Deep learning for computer vision: A brief review”. In: *Computational intelligence and neuroscience* 2018 (2018).

[WG19a] William Whiteley and Jens Gregor. "CNN-based PET sinogram repair to mitigate defective block detectors". In: *Physics in Medicine & Biology* 64.23 (2019), p. 235017.

[WG19b] William Whiteley and Jens Gregor. "Direct image reconstruction from raw measurement data using an encoding transform refinement-and-scaling neural network". In: *15th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*. Vol. 11072. International Society for Optics and Photonics. 2019, p. 1107225.

[WYDM20] Ge Wang, Jong Chul Ye, and Bruno De Man. "Deep learning for tomographic image reconstruction". In: *Nature Machine Intelligence* 2.12 (2020), pp. 737–748.

[Xie+19] Zhaoheng Xie et al. "Generative adversarial networks based regularized image reconstruction for PET". In: *15th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*. Vol. 11072. International Society for Optics and Photonics. 2019, 110720P.

[Yan+18] Qingsong Yang et al. "Low-dose CT image denoising using a generative adversarial network with Wasserstein distance and perceptual loss". In: *IEEE Transactions on Radiation and Plasma Medical Sciences* 37.6 (2018), pp. 1348–1357.

[YCH21] Hanene Ben Yedder, Ben Cardoen, and Ghassan Hamarneh. "Deep learning for biomedical image reconstruction: A survey". In: *Artificial Intelligence Review* 54.1 (2021), pp. 215–251.

[Ye+18] Dong Hye Ye et al. "Deep back projection for sparse-view CT reconstruction". In: *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE. 2018, pp. 1–5.

[ZD20] Hai-Miao Zhang and Bin Dong. "A review on deep learning in medical image reconstruction". In: *Journal of the Operations Research Society of China* (2020), pp. 1–30.

[Zha+18] Zhicheng Zhang et al. "A sparse-view CT reconstruction method based on combination of DenseNet and deconvolution". In: *IEEE Transactions on Medical Imaging* 37.6 (2018), pp. 1407–1417.

[Zhu+17] Jun-Yan Zhu et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2223–2232.

[Zhu+18] Bo Zhu et al. "Image reconstruction by domain-transform manifold learning". In: *Nature* 555.7697 (2018), p. 487.