Medical Imaging techniques for Transformation and Inference using Deep Learning

A Thesis Submitted

in Accordance with the Requirements

for the Degree of

MASTERS (By Research)

By

PREETHI SRINIVASAN

(S18001)

to the

SCHOOL OF COMPUTING AND ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY MANDI

February, 2021

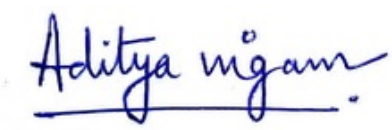
DECLARATION

This is to certify that the Thesis entitled "Medical Imaging techniques for Transformation and Inference using Deep Learning", submitted by me to the Indian Institute of Technology Mandi for the award of the Degree of Master of Science [by Research] is a bonafide record of research work carried out by me under the supervision of (Dr. Aditya Nigam and Dr. Arnav Bhavsar). The content of this Thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any Degree or Diploma.

Mandi, 175 005 Preethi Srinivasan

CERTIFICATE

It is certified that the work contained in the thesis entitled "Medical Imaging techniques for Transformation and Inference using Deep Learning" by Preethi Srinivasan, for the award of the degree of MS (by Research) has been carried out under our supervision. The results embodied in this thesis, in full or in parts have not been submitted elsewhere for any degree or diploma.



Dr. Aditya Nigam (Advisor)
Assistant Professor
School of Computing and Electrical Engineering
IIT Mandi, H.P., India

Dr. Arnav Bhavsar (Co-Advisor)
Assistant Professor
School of Computing and Electrical Engineering
IIT Mandi, H.P., India

Feb, 2021

ABSTRACT

Medical imaging has significantly progressed to yeild high quality visual representations of the organs inside the body and is of critical value to health care. Multiple imaging modalities such as MRI, X-ray, CT and Ultrasound exist to serve different diagnostic purposes. Nevertheless, in addition to the actual energy signal, post-processing approaches exists which are intended to assist the diagnosis by performing simple improvements like enhancing the sharpness and reducing image noise, providing intelligent suggestions by segmenting the artefacts, classifying the diseases, making meaningful and critical inferences and enabling mass screening. These post-processing tasks can be improved and yield benefits such as decreased acquisition time, cost, need for expert training, increased comfort, and decreased radiation hazard. In this thesis, we have explored deep learning-based techniques for some advanced post-processing tasks like synthesise MR images, automate X-ray report generation, and denoise the CT Scan.

Synthesising inter modality images of MRI: MRI imaging can be utilised to interpret the distinct nature of tissues, characterised by two relaxation times, namely T1 and T2, producing contrasting yet related information. In order to reduce the acquisition time and thereby alleviate comfort and reduce the per-person cost, we propose an Encoder-Decoder-based deep learning architecture to reconstruct T2 weighted image from T1 weighted image.

Automating X-ray report generation: We propose an attention-based deep neural network to generate X-ray report automatically. X-rays can be used for mass screening in several critical/pandemic scenarios as is fast and cost effective.

Denoising low dose CT Scan: Computed Tomography (CT) scanners induce X-ray radiation through the body to capture images of the bones and tissues. A higher radiation dosage leads to clearer images but have harmful effects. We propose an architecture that computes visual attention across non-overlapping patches to denoise the low dose CT scans.

Dedicated

to

Gautam

Acknowledgements

I would like to thank my advisors Dr. Aditya Nigam and Dr. Arnav Bhavsar, for supporting me during these past three years. I am deeply grateful to Aditya sir for patiently nurturing me in the beginning and constantly encouraging me to push my limits. He inspires me to learn continuously and listen first. I am indebted for his patience to rigorously review and give feedback for the manuscripts during my research. I am very thankful to Arnav sir for all the insightful discussions. He implicitly fed my problem solving skill by asking the right questions during my research. He is also one of the coolest and smartest people to hang out with. My sincere thanks to the advisory committee for their consistent support.

I am very fortunate and grateful to have a loving family who have supported me unconditionally. A good peer group is important to stay motivated. I am lucky to find a wonderful friend and a mentor in Daksh. I had an enriching time working and learning with him. My first project was with Prabhjot, and she is one of the highly optimistic and motivated people I know. I am blessed to have spent time with some beautiful people like Mohana, Avantika, Ranjeet, Jyoti mam, Anshul, Sujeet sir, Krati, Ganesan, Merlin, Krishan, Arshdeep, Dikshita, Isha and Deepak.

Preethi Srinivasan

Contents

List of Figures

List of Tables

List of Abbreviations

ANN Artificial neural network

DL Deep Learning

ML Machine Learning

MRI Magnetic Resonance Imaging

T1WI T1 Weighted Image

T2WI T2 Weighted Image

CSF Cerbero Spinall Fluid

TE Echo Time

TR Repetition Time

RF Radio Frequency

HCP Human Connectome Project

PSNR Peak Signal to Noise Ratio

MSE Mean Square Error

SSIM Structural Similarity Index Measure

BET Brain Extraction Tool

FLIRT FMRIB's Linear Image Registration Tool

MTI Medical Text Indexer

GAP Global Average Pooling

BLEU Bilingual Evaluation Understudy

SOTA State of the Art

CT Computed Tomography

kV kilo Volts

mA milli Amperes

ALRP As Low as Reasonably Achievable.

LDCT Low Dose CT

FDCT Full Dose CT

FBP Filtered Back Projection

IR Iterative Reconstruction

TCIA The Cancer Imaging Archive

NIH National Institute of Health

PGIMER Postgraduate Institute of Medical Education and Research

GPU Graphic Processing Unit

GAN Generative Adversarial Network

MLP Multi Layer Perceptron

Chapter 1

Introduction

1.1 Medical Imaging

Arguably, most medical issues occur inside the body, and it is often hard to look at the person and figure out the problem simply from the visible clinical symptoms. Thus, many Computer-Aided-Diagnosis (CAD) based medical imaging systems provide a visual representation of the organs inside the body. This has had a significant effect on making an improved diagnosis. Among the variety of medical imaging techniques, the common ones include radiography, computed tomography (CT), magnetic resonance imaging (MRI), nuclear medicine and ultrasound Figure 1.1 which are briefly discussed below.

A: Radiography uses electromagnetic radiation to take images of the inside of the body. The most well-known and common form of radiography is X-ray imaging. For this procedure, an X-ray machine transmits high-energy waves onto the body. While the soft tissues, like skin, do not absorb these waves, the hard tissues like bones absorb them. The machine projects the results onto a film creating 2D images. Areas with high levels of calcium (bones and teeth) block the radiation, causing them to appear white on the X-ray image. Soft tissues allow the radiation to pass through. As a result, they appear grey or

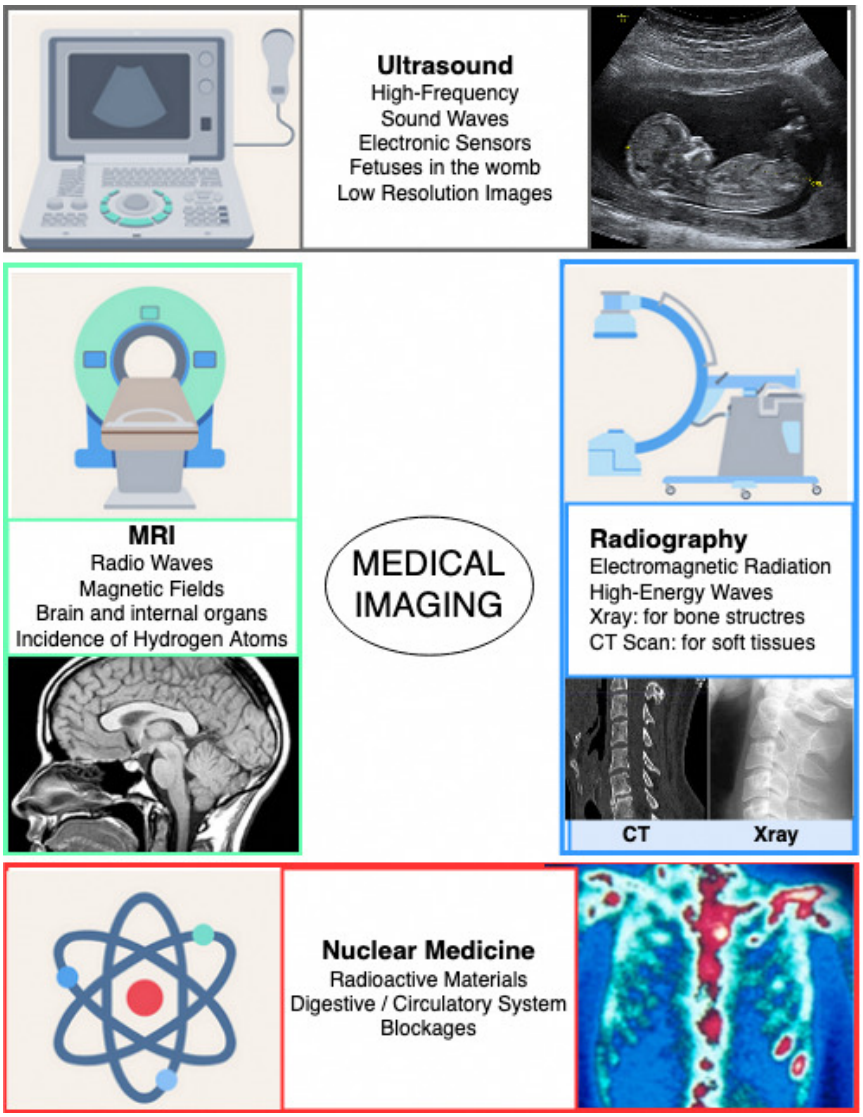


Figure 1.1: Broad classification of major Medical Imaging techniques and some key pointers.

black on the image. X-Ray is good for identifying fractures, dislocations and misalignment. However, it won't show subtle bone injuries, soft tissue injuries or inflammation.

B: CT Scan uses the same electromagnetic rays like X-ray imaging to picture soft tissues, hard tissues as well as blood vessels at the same time. However, unlike a simple X-ray study, it offers a much higher level of detail, creating computerized 3D images, giving a 360-degree view and thus making it a more powerful form of X-ray. The patient is slid

under the tunnel-like structure and the machine, equipped with X-ray sensors and receivers, rotates around to produce cross-section images of the body. The images of internal organs are very detailed and allow doctors to make decisions on the most accurate treatment plan to take. CT imaging takes longer than X-rays but is still fast and takes about a minute to acquire the image. This makes it desirable during emergency conditions. In addition, CT Scans can spot blood clots, cancer, tumours, subtle bone fractures invisible in a X-ray, etc,.

C: Magnetic Resonance Imaging employs a powerful magnetic field B_0 to force the net magnetic moment (M) of protons (hydrogen atoms) in the body to align with B_0. Then, a radio frequency (RF) pulse is applied through the patient causing M to move away from B_0. Then, when the radio frequency pulse is turned off, M tries to realign with B_0, and in the process of realigning themselves to the magnetic field B_0, protons emit energy which the MRI sensors capture. The time it takes for the protons to realign with the magnetic field (also called the relaxation time) and the amount of energy released change depending on the chemical nature of the tissue molecules. This phenomenon leads to the contrast in the image.

Realignment occurs in two directions, in longitudinal and transverse directions, which causes T1 decay and T2 decay mechanisms, respectively. Image contrast can be controlled by two parameters called TR (Repetition time) and TE (echo time). TR refers to the time duration between the application of RF excitation pulse and the next pulse, and TE refers to the time duration between the application of RF excitation pulse and the peak of the echo detected. TR and TE provide different levels of sensitivity to differences in relaxation time between various tissues. For example, at short TRs, difference between relaxation time of fat tissues and water can be detected because longitudinal magnetization recovers faster in fat than in water. So TR is related to T1 decay and therefore affects the image

contrast in T1 weighted image. Similarly, at long TEs, transverse magnetization's recovery leads to differences in relaxation time of fat tissues and water. Hence TE is related to T2 decay and therefore affects image contrast of T2 weighted image [4].

MRI scanners can acquire high-resolution images of the brain and other internal organs. Unlike X-rays and CT Scans, MRI does not use any ionizing radiation. MRI is a widely used imaging modality due to its non-invasive nature and ability to provide distinct characteristics of soft tissues across different modalities such as T1, T2, FLAIR, proton density (PD), functional-MRI and diffusion-MRI. In addition, it is useful to spot musculoskeletal conditions such as cartilage loss, joint inflammation, nerve compression, spinal injuries, torn or detached ligaments.

D: Nuclear Medicine is a general term that involves any medical use of radioactive materials. However, in terms of imaging, it usually refers to the use of radioactive tracers, which are radioactive materials that are injected or swallowed so that they can travel through the digestive or circulatory system. The radiation produced by the material can then be detected to create an image of those systems. Nuclear medicine is used when you need to look inside the digestive or circulatory systems, such as for blockages. Positron Emission Tomography or PET is the most common imaging modality in nuclear medicine. Nuclear Medicine is used to assess the function of an internal organ, while the CT Scan is used to look at the internal organ.

E: Ultrasound utilizes high-frequency sound waves, which are reflected off tissue to create images of organs, muscles, joints, and other soft tissues. The ultrasound machine transmits the sound pulses using a probe; the waves travel till they hit a tissue, and the reflected waves are picked up by the probe and relayed to the machine. The machine calculates the distance from the probe to the tissue based on the speed of the sound in the tissue, and the time of each echo's return indicates the intensity. These distances

and intensities form a two-dimensional image on the screen. Ultrasound is used to look at fetuses in the womb and take images of internal organs when high resolution is unnecessary.

1.1.1 Medical Image Processing

Data acquisition from the patient gives raw data. Typically a raw digital image goes through several processing steps before it is made available to a human observer. This can be divided into two parts - **a) pre-processing** and **b) post-processing**. As shown in the Figure 1.2, as an example for MRI data, preprocessing reconstructs an image from the raw data (kspace data in case of MR image), while post processing is intended to assist the diagnosis - from performing basic improvements like changing the image contrast, enhancing the sharpness and reducing image noise, to giving intelligent suggestions by segmenting the regions of interest, classifying the diseases and to even making semantic level inferences from a medical image.

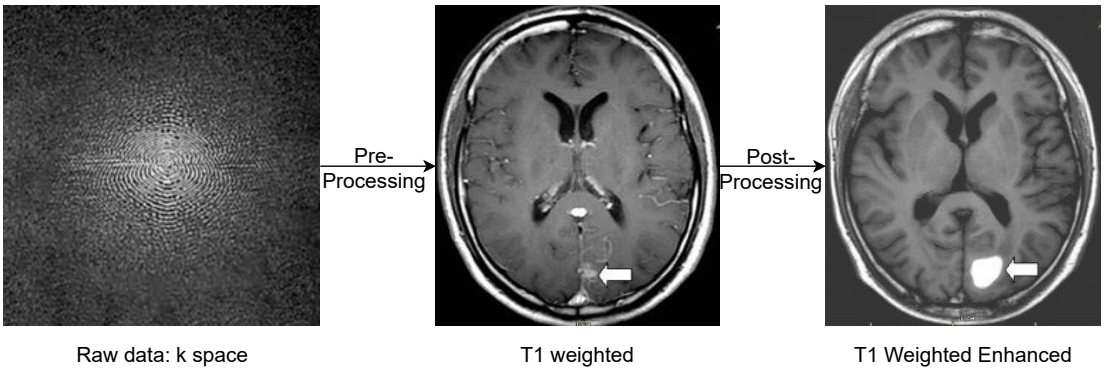


Figure 1.2: Stages of Pre and Post processing. k space is the raw data format of MRI in frequency domain. During preprocessing k space is converted to T1 weighted image which is one of image modalities of MRI in spatial domain. It is enhanced with post processing algorithms for better contrast.

Typically the range of post processing problems can be categorised into the following buckets:

- Image Analysis

- Image Enhancement
- Image Restoration
- Image Compression
- Image Synthesis

While traditionally post-processing involved simple image processing methods, learning-based methods, which yield superior results, are being reported in recent years. Computational algorithms can give a preliminary opinion based on objective analysis after looking at thousands of globally available data samples. Model-based approaches usually use implicit or explicit a-priori knowledge of the shapes and appearance of relevant structures into the process or involve simplistic approximations to model how the image is formed. However, biological structures are inherently subject to inter and intra variability from patient to patient and sometimes within the same patient. Due to the heterogeneous (unstructured/incomplete/irregular) nature of the images, formulations based on such assumptions may be inaccurate. This system is influenced by a set of predetermined features computed in a predetermined manner.

1.1.2 Impact of Deep Learning

Features can represent variations in pixel values, shape, textures, position or/and orientation. The performance of most of the traditional machine learning algorithms depends on how relevant such features are. From the modelling perspective too, the traditional approaches are arguably based on some assumptions. For instance, generally, image enhancement problems can be formulated as x = Hy + e where “x” is the input, “y” is the output, “H” is the transformation matrix and “e” gives in for generalization error. This is an ill posed problem that one is trying to solve. Unlike the above model, the transformation between input and output can be highly non linear in the real world scenario because of the spatial variations and complex details in the medical image. Some examples of such

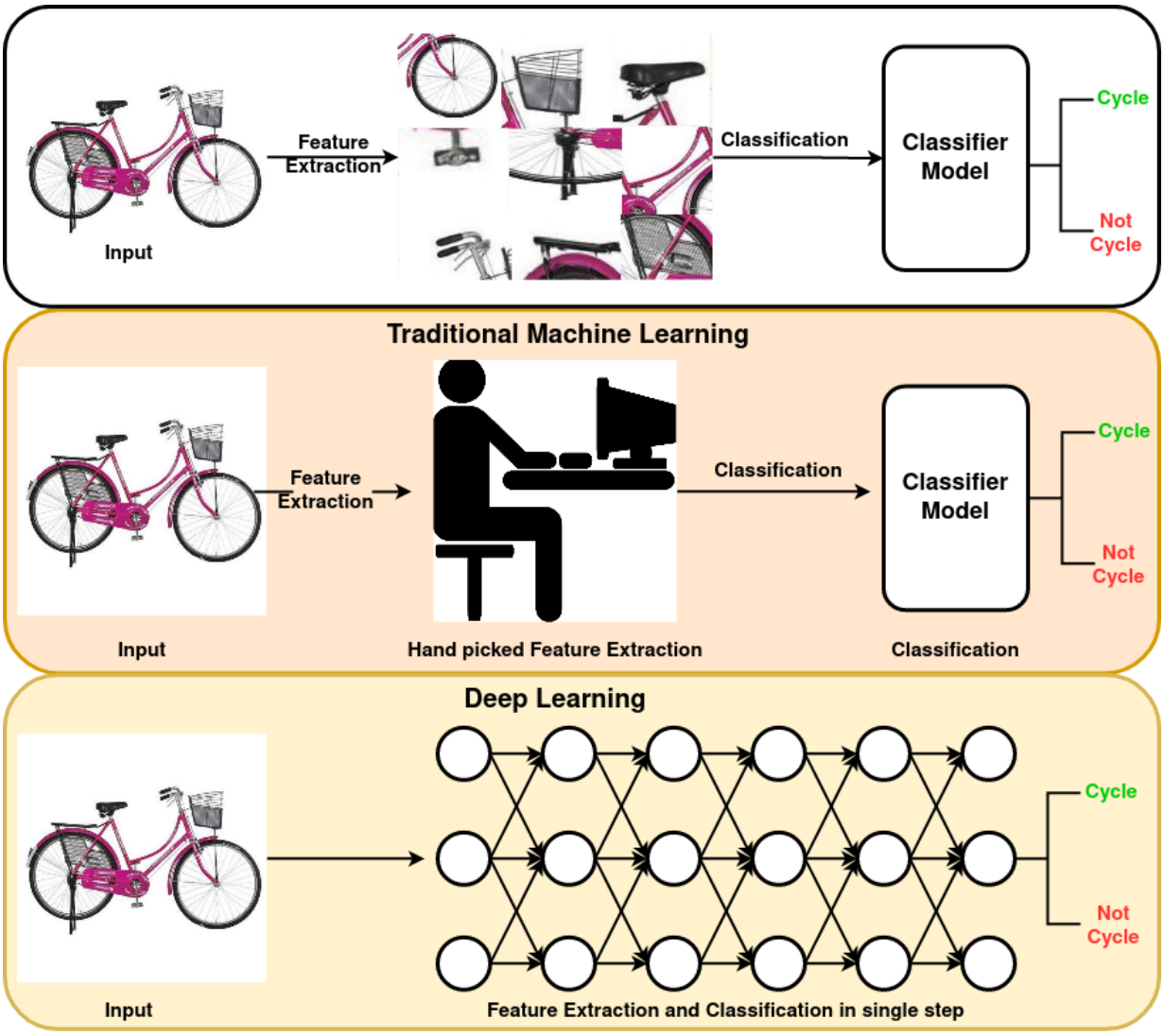


Figure 1.3: Feature Extraction step picks all the discriminating features, Classification step determines the class of the output, Traditional learning algorithm features are hand picked separately as a first step before classification. Deep Learning merges both the feature extraction and classification step.

traditional model-based methods involve Compressed sensing reconstruction techniques, Bayesian Markov random field models, Rough set theory and Higher order PDEs.

On the contrary, deep learning models can learn the features from the data as shown in Figure 1.3. Extracting features and solving the problem are merged into one step. Abundance of digital data of medical images available can improve the performance of DL

algorithms without having the need to know much a-priori knowledge.

The progress in diagnosis is revolutionary because of the engineering solutions and has particularly been drastic since the evolution of Deep Learning. Deep neural networks are now the state-of-the-art machine learning models across a variety of areas, from image analysis to natural language processing, and widely deployed in academia and industry. These developments have a huge potential for medical imaging technology, medical data analysis, medical diagnostics and healthcare in general. Application in one such pipeline is discussed here in Table 1.1. Even though there is commendable progress in this area, every medical image modality has certain trade-offs as shown in Table 1.2. There is ample scope for improvement to mitigate the cons mentioned in the above table.

1.2 Motivation of the Thesis

Typically the metrics used to judge a modality are the following:

- Acquisition Time (AQT): Time spent by the patient for the acquisition of the medical image. This time includes a) Participating in the procedure of image acquisition and b) Post-processing.
- Comfort (CMT): Comfort levels of the patient during the procedure. Most of the imaging procedures require the patient to stay still. Comfort may also relate to the long distances required to travel in Tier-2 towns to access diagnostic centres with Radiologists.
- Cost (CST): Cost of the procedure.
- Number of Radiologists (NoR): Availability of radiologists at the clinics, especially in Tier-2 cities.

Application Area	**Popular DL Based Solution**
Medical Image Segmentation	Segmenting the region of interests is a holy grail to the radiologists for assessment of the disease. UNet [5] containining symmetrical Encoder-Decoder architecture with skip connections is among the most famous network architectures for segmentation. Lately Transformers are used by TransUNet [6] for medical image segmentation.
Medical Image Reconstruction	Novel generative networks can reconstruct images on the fly. [7] is a GAN-based reconstruction method with noisy and/or incomplete measurements. It demonstrated that even fine structures could be recovered in the object relevant for the medical diagnosis that may be difficult to achieve using traditional reconstruction methods relying on sparsity-promoting penalties.
Medical Image Restoration	Denoising and artefact detection are outperformed by state of the art deep convolution-based Encoder-Decoder networks such as [8]. Attention-based neural networks such as [9] also denoise the 3d volume acquisitions of radiology images like CT Scan.
Medical Image Super-resolution	Reconstructing high-resolution images from low-resolution images is a prominent application of DL in medical imaging because high-resolution images are either associated with heavier amounts of the stimulant as compared to low-resolution images or high cost. Wide range methods have been explored using deep neural networks as in [10], CNNs, Attention-based learning [9] and [11] uses GAN based enhancement technique for cytopathological images.
Medical Image Synthesis	Synthesis in medical imaging usually refers to generating images with new parameters or tissue contrast for faster imaging. This also helps in creating a huge reservoir of training data. Unsupervised synthesis is a growing area with a lot of demand for many practical purposes. Many successful algorithms in this area is based on GAN [12] [13]. [14] is another latest self supervised method.
Medical Image Registration	Scan volumes are estimated from multiple images of different scanners, light exposures, subject to movement artefacts etc,. Registering them into one generalized form is a herculean task given the inter and intra variability in patients. [15] and [16]summarizes the latest techniques using DL.
Medical Image Analysis	Inferring the diagnosis from medical image as human doctor would do is a newly emerging field. [17] is a self-supervised framework for disease classification and segmentation. [18] is a survey on techniques related to keeping a human in the loop during medical image analysis.

Table 1.1: Application Areas of DL in Medical Imaging

Imaging Modality	**Pros**	**Cons**
Ultrasound: High frequency sound waves	Clear visibility of internal organs like fetuses in womb	Low Resolution Images. Cannot capture bone structures. Requires expert training.
Nuclear Medicine: Radioactive materials.	Identify blockages in digestive system.	Very high operational costs. Exposure to harmful radiations. Requires expert training.
Radiography: X-ray.	Can observe bone structures and opacity in lungs.	Requires expert training.
Radiography: CT Scan.	Detects soft tissues.	Exposure to harmful radiations.
MRI: Magnetic fields.	Imaging brain and internal organs clearly.	Very high operational costs. Undetected metals implants can be exposed to high magnetic field. Can cause claustrophobia. Requires expert training.

Table 1.2: Pros and Cons of Imaging Modalities.

- Radiation Hazard (RHD): Exposure to harmful radiation contributes to an increased risk of cancer in the population.
- Exposure to Closed Environment (ECE): The patient is required to spend over 45 mins in a claustrophobic environment for an MRI scan.
- Need for Expert Training (ETG): Amount of training required to acquire the skill to read and diagnose the medical images.
- Quality (QLT): High-resolution images are of higher quality and help accurate diagnosis. It is usually accompanied by high radiation exposure.

In this thesis, we have tried to answer the question **"Can we do better?"** for each problem in terms of all the relevant metrics described above - AQT, CMT, CST, NoR,

RHD, ECE, ETG and QLT. We propose deep learning-based post-processing algorithms to mitigate the cons and improve the metrics mentioned above on three specific domains: MRI, X-ray and CT Scan. Issues and motivation for the three problems are as follows:

Problem 01: Long acquisition time (AQT) due to sequential acquisition of multi-modality MR images, especially T2 weighted images (T2WI) with longer AQT acquired after T1 weighted images (T1WI), though beneficial for disease diagnosis, is practically undesirable. Can the patient spend less time in the diagnostic centre? In imaging techniques such as MRI, the patient is subjected to closed units for a long time (45 minutes in the worst case) and is required to be steady throughout the procedure, which becomes challenging in the case of children and older people. MRI is very expensive at about 10000 for a scan. Less time per patient also means, lower price to value ratio, which eventually cuts costs for the patient. See Figure 1.4 for figurative demonstration of the same.

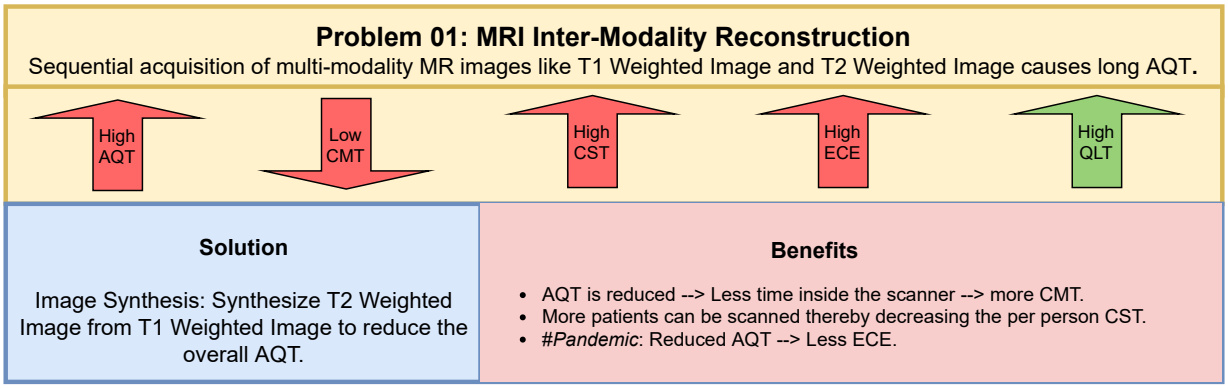


Figure 1.4: Motivation of Problem 01. Direction of arrow describes High/Low quantity. Green arrow indicates desirable quality about the image modality. High AQT leads to low CMT, high CST and high ECE as shown by the red arrows indicating un-desirable quality.

Problem 02: Can we generate a preliminary report without the need for the presence of a radiologist? General physicians are not trained to read medical images such as X-ray. They rely on radiologists to make inferences before treating the patient. The absence of radiologists in the clinics can lead to bottlenecks. Also, the Doctor to patient ratio is

extremely skewed in India overall, particularly in Tier 2 and Tier 3 locations. Can we have a cheaper and preliminary report generated before the patient can travel several 10s of kilometres for simple procedures? With the internet invasion to every nook and corner of the country, physicians can also be consulted online with this automated report. See Figure 1.5 for figurative demonstration of the same.

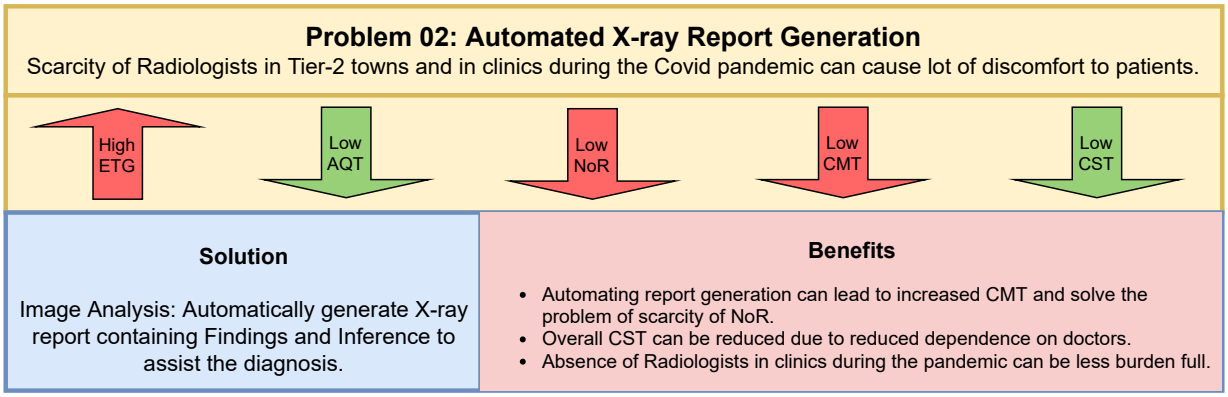


Figure 1.5: Motivation of Problem 02. Direction of arrow describes High/Low quantity. Green arrow indicates desirable quality and Red arrow indicates the un-desirable quality about the image modality.

Problem 03: Can we make these imaging procedures less invasive? Computed Tomography (CT) scanners induce X-rays through the body to capture images of the bones and tissues. Higher dosage of radiation leads to clearer images compared to Low dose images. Nevertheless, higher dosages of radiation have harmful effects. Alternatively, MRI can do the same job at a much higher cost and time inside the scanner leading to discomfort. MRI scanners are not affordable in smaller places in developing nations. Furthermore, we proposed techniques to address the availability of only fewer or no paired data to learn the image transformation function. See Figure 1.6 for figurative demonstration of the same.

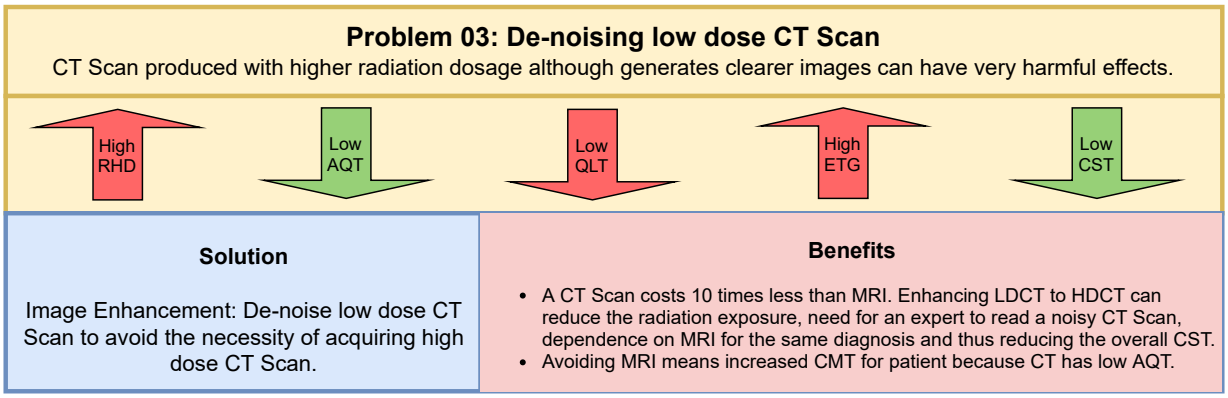


Figure 1.6: Motivation of Problem 03. Direction of arrow describes High/Low quantity. Green arrow indicates desirable quality and Red arrow indicates the un-desirable quality about the image modality.

1.3 Thesis Contribution

We have proposed deep learning based models that can lead to some progress regarding the questions asked above in three chosen areas. See below for a succinct explanation of the problems we solved.

- **MRI Inter-Modality Reconstruction:** We propose an image synthesis technique using an encoder-decoder architecture to reconstruct the T2WI from T1WI. We also explored a scenario of reconstructing T2WI in the presence of under sampled T2WI along with T1WI.

- **Automated X-ray Report Generation:** Automatically generating report from X-ray is a type of Image Analysis. Correlation between the image features and sequential textual information regarding the X-ray is learnt from the training data to predict the report of unseen Xrays. We proposed an attention-based mechanism to learn the salient features from each patient's X-ray image and tags.

- **De-noising low-dose CT Scans:** Data available for research from a hospital is usually available in unpaired format unlike the ones in research consortium. Its

practically not feasible to procure both low dose scan and high dose scan for every patient. We propose a image enhancement solution in a semi supervised set up to de-noise low dose CT scans for easy diagnosis.

1.4 Organization of Thesis

This thesis starts with an Introduction chapter highlighting the area of medical image post-processing, the pros and cons of solutions before Deep Learning, the impact of Deep learning, motivation of this work and a brief description of the proposed work. Chapter 2 covers all the preliminaries and theory in a digest that is required to understand the sub modules of proposed algorithms. Chapter 3 discusses the image synthesis problem using MRI. Chapter 4 involves the image analysis part with the automated report generation process. Chapter 5 deals with the image enhancement part concerning CT Scans. Chapter 6 summarizes the three problems tackled during the research and discusses the possible future areas that can be explored in alignment with this work.

Chapter 2

Preliminaries

This chapter aims is to discuss the preliminary concepts that can be used in the rest of the chapters. Therefore this chapter begins with an overview of fundamental components of deep learning architectures. It is followed by introducing autoencoder, metric learning, properties of transformer architecture and a brief description about Cycle GAN to set a foundation for the upcoming chapters, which discuss deep learning-based techniques on medical images.

2.1 Overview of a Deep Learning architecture

The notion of *Learning* is succinctly defined by Mitchel (1997) [19] by stating that "A computer program is said to learn from experience $[E]$ with respect to some class of tasks $[T]$ and performance measure $[P]$, if its performance at tasks in T, as measured by P, improves with experience E" as shown Figure 2.1. Deep learning is a particular kind of machine learning paradigm that achieves by learning as a nested hierarchy of functions. Each function is defined in simpler functions, and abstract representations.

Neural networks is made up of artificial units called neuron. A biological neuron

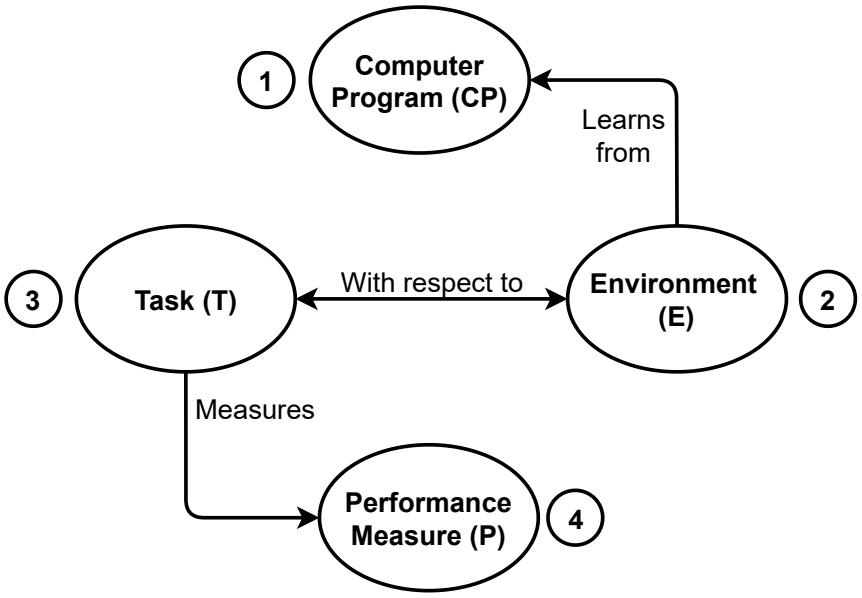


Figure 2.1: Definition of Learning

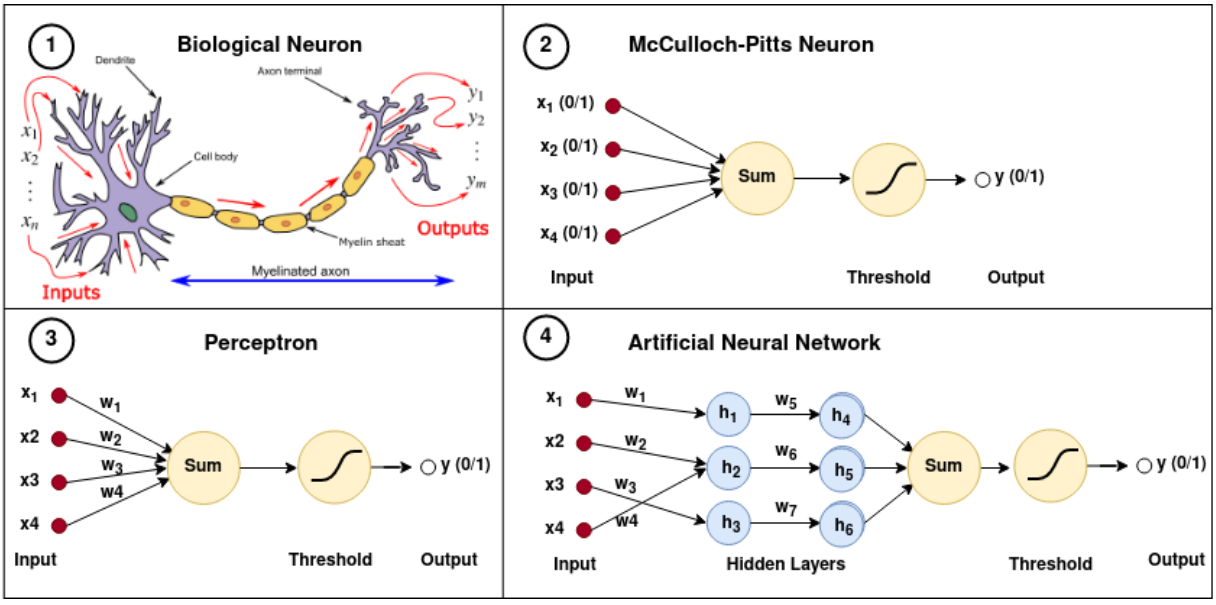


Figure 2.2: Evolution of Artificial neural networks from : 1. Biological neuron to 2. Simple aggregate of Boolean values with McCulloch-Pitts neuron to 3. Weighted aggregate of Real values with Perceptron and 4. Deeper network with hidden layers.

inspired McCulloch and Pitts to model an artificial neuron called the **McCulloch-Pitts** neuron [20] in 1943, which takes a binary input, aggregates the input values and produces 1 or 0 depending on the threshold. Thresholding at the output acts as the non-linearity

component in the network. Later in 1957, Frank Rosenblatt developed a slightly tweaked version called **Perceptron** [21] where the neuron accepts real values, and the output is the weighted sum of the input values. Each of these fundamental units is called a neuron and a layer in a neural network can be composed of several of these neurons. These neurons communicating with each other across layers form an Artificial Neural Network. More the number of layers deeper the network. Layers between the input and output are called the hidden layers. This idea forms the basis for the Deep Neural Networks as shown in Figure 2.2.

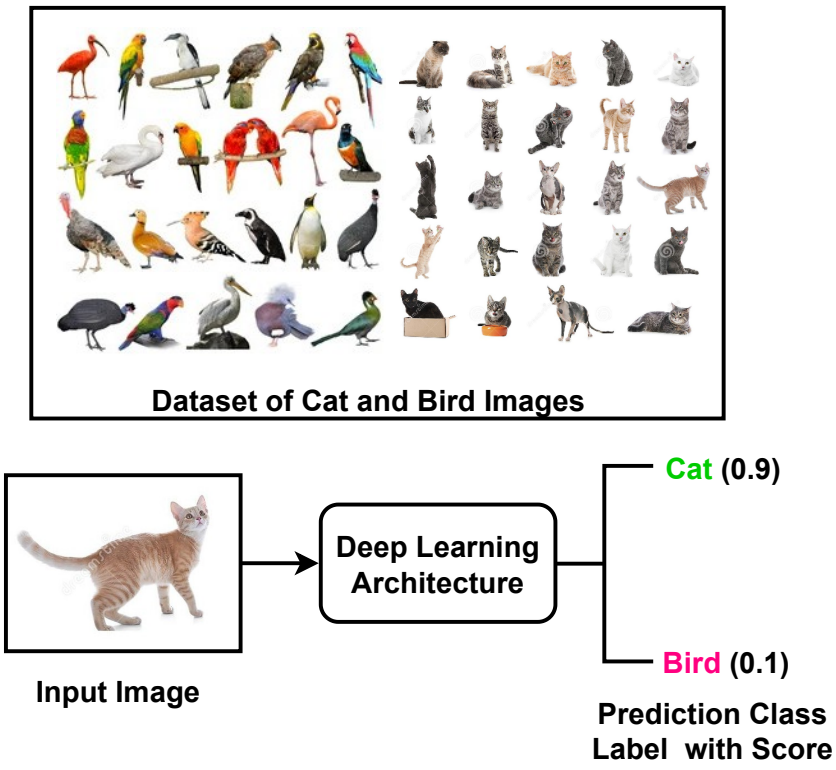


Figure 2.3: Demonstration of Classification Task

Deep Learning networks have established the superiority, initially, in solving the classification task where the model tries infer a categorical label from the observed input. The classification problem categorizes the input into one or more classes. The problem is solved in a supervised fashion if both input and the corresponding output labels are provided to

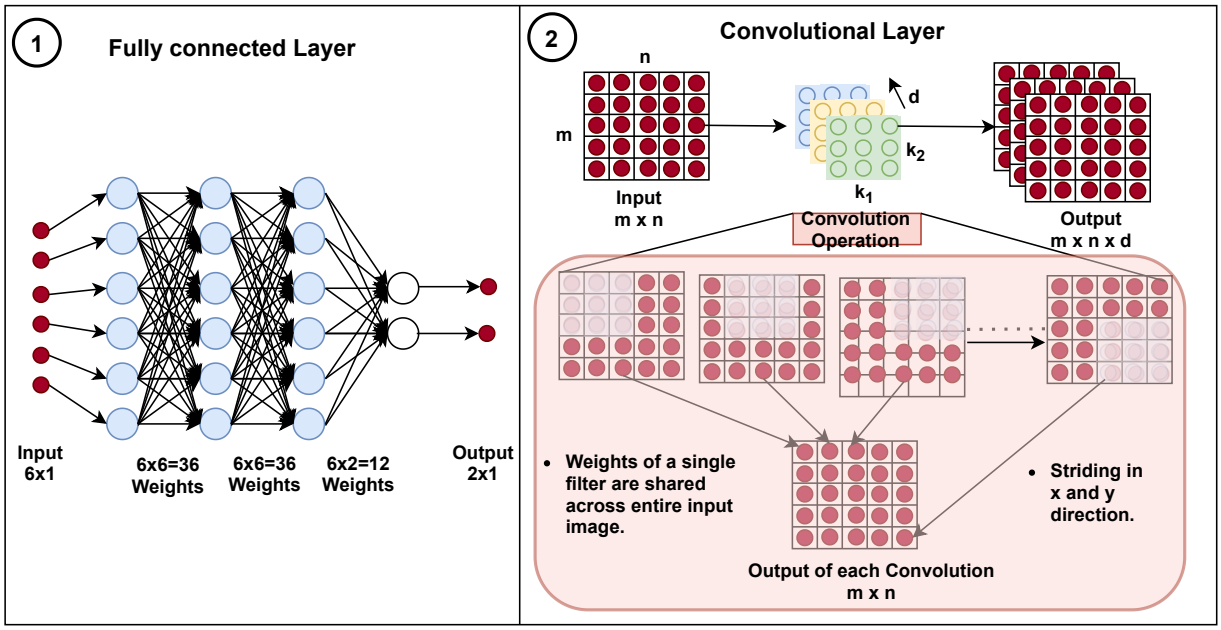


Figure 2.4: Types of Layers. 1: Fully Connected Layer and 2: Convolutional Layer: On single channel input, three different filters represented with three different colors are applied to produce three outputs, respectively. Convolution operation with a single filter, stride size one and padding is explained in detail in the shaded region. The element-wise multiplication of the filter and patch is followed by summing the values to produce a scalar value.

learn a neural network. For example, consider two classes of images - Cats and Birds as shown in Figure 2.3 and the task is to train a Deep Neural Network so as to learn to predict the class of a given image at the test time. The task comprises learning the discriminating features of the image, and use these to classify the image into one of the classes. The rest of this section discusses basic concepts of building a Deep Learning architecture with respect to the above classification problem.

2.1.1 Different Types of Layers

One can observe from Figure 2.4 that various layers have been devised depending on the density of connections, weights used in computation between the inputs and neurons. Each of these following layers is used for different kinds of tasks, and input to each hidden layer

and the final layer is the output from the previous layer.

Fully Connected Layer (FCL): It has all the neurons connected to each neuron of the previous layer as shown in (1) of Figure 2.4. Each of these connections carries a certain weight which is learnt during the training of the network. The number of connections between any two layers is the product of the number of neurons in both layers. FCL is mostly used in the final layers of the classification task. FCL is a replication of matrix multiplication, and they are structure agnostic. They do not assume any structural information in the input. If the input is an image where a spatial relationship exists, FCL may not learn the right features, and it is also computationally very intensive. Then convolutional layer came to the rescue.

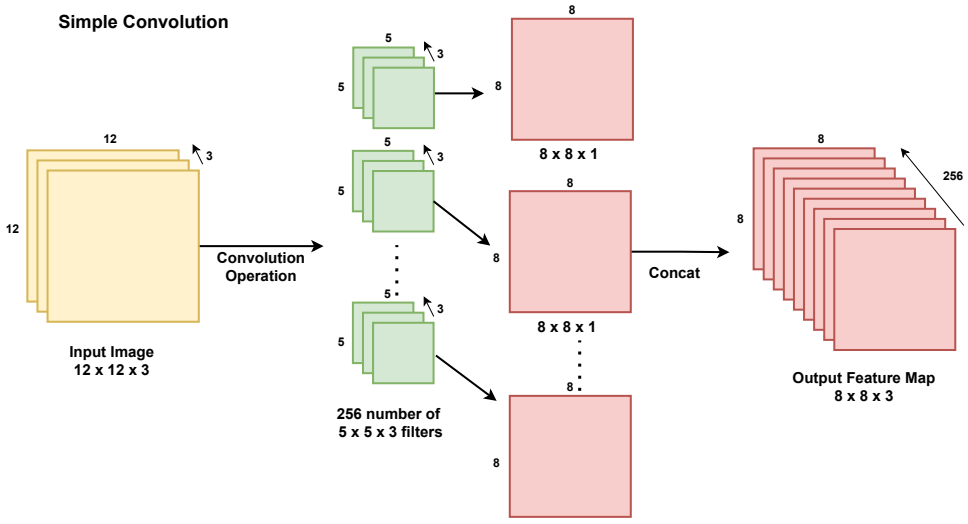


Figure 2.5: Simple Convolution Operation: Number of computations and parameters

Convolutional Layer (CL) As shown in the 2nd part of Figure 2.4, assuming the input is an image of size 5x5 pixels, convolution on the image with a 3x3 filter upon padding will produce a 5x5 output. The number of outputs is equal to the number of filters. Over a multi-channel input, filters implicitly take the 3rd dimension of the size equal to the number of channels in the input in case of a 2-D convolutional operation. Similarly, a 3-D

convolutional operation is applied on the input of size 4-dimension, where each channel is of 3-dimensional size. Here, filters implicitly take the 4th dimension of the size equal to the number of channels in the input.The training in the convolutional neural network (CNN) is essentially the process of learning these filters. Output of each filter is referred to as a Feature Map. Multiple filters are required to observe the input data in different ways that may be useful for other objectives. As with Figure 2.3, each filter is expected to extract such features from the cat image that can be useful to discriminate against a bird image. For instance, identifying a tail in a cat image or a beak/wings in a bird image can be useful for the appropriate classification. In a CNN, filters in the initial layers are known to learn low-level features like edges, and deeper layer filters learn high-level features like the overall face texture of the cat image as in Figure 2.3.

Images typically have a property called stationarity which states that a pattern in a certain location on the image is very likely to repeat in other locations of the image. So a single filter strides across the entire image to exploit the property of stationarity in the images. This property is also called weight sharing. Irrespective of which corner the cat is located in the input image in Figure 2.3, a learned filter can extract the relevant features. Lets calculate the number of computations required in the convolution operation: On a $12 * 12 * 3$ input image as shown in Figure 2.5, with a $5 * 5 * 3$ filter (channel size of filter is implicitly equal to channel size of the input) and 256 such filters will give rise to $(256)*(5*5*3)*(8*8) = 1,228,800$ computations and $(256)*(5*5*3) = 19,200$ parameters. With strides of size 1, the filter moves $8 * 8$ times along all the rows and columns over the input image, thus reducing the output image size after convolution operation to $8*8*1$. If the input image is padded with 5 rows and 5 columns, the filter moves $12*12$ times over the input image and produces an output image of the same height and width $12 * 12 * 1$. Since the filter moves only in the x and y direction along the rows and cols, this operation is called 2D-Convolution. Figure 2.4(2) figuratively shows the operation on a $5*5$ size of input with

$3 * 3$ size of filter. The number of parameters and computations becomes exponentially large for a bigger input image, and real-world images are at least over $200 * 200$ pixels. Performing convolution to utilize the image properties while also decreasing the number of computations and parameters extracts the image features efficiently.

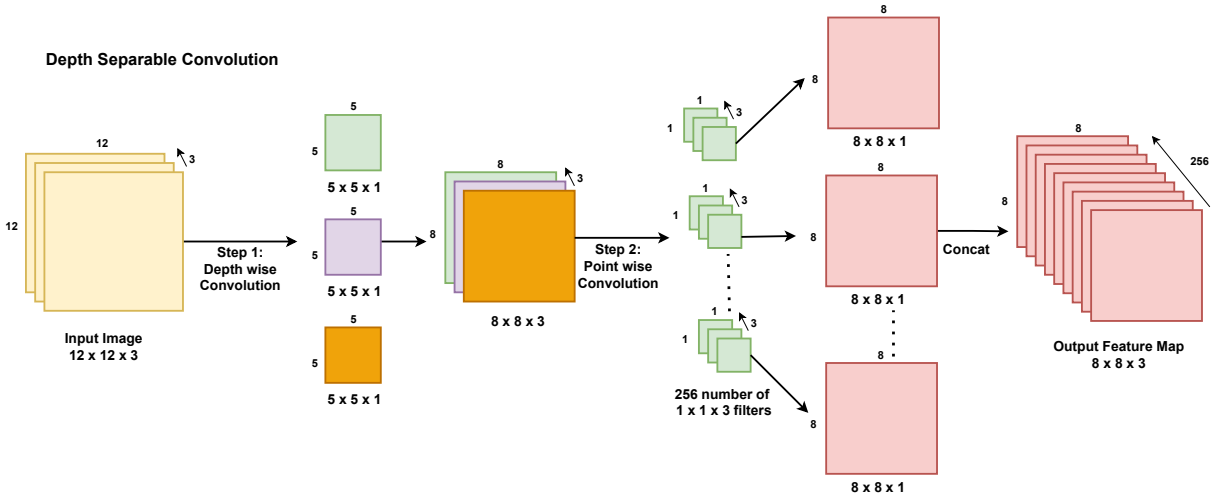


Figure 2.6: Depth Separable Convolution Operation: Number of computations and parameters

Depth Separable Convolution [22] on the other hand, as shown in Figure 2.6 drastically decreases the number of computations by separating the above operation into two steps.

- Step 1: Three separate $5 * 5$ filters are applied on $12 * 12 * 3$ input image to produce $8 * 8 * 3$ output.
- Step 2: 256 number of $1 * 1 * 3$ point wise convolution operation is performed on the intermediate output to produce final feature maps of size $8 * 8 * 256$.

The number of computations in this case are $3 * 5 * 5 * 8 * 8 = 4,800$ from Step 1 and $256 * 1 * 1 * 3 * 8 * 8 = 49,152$ from Step 2 summing up to $53,952$ and the number of learnable parameters from both the steps sum up to $5 * 5 * 3 + 1 * 1 * 3 * 256 = 843$. The number of computations and parameters in a depth-separable convolution operation is reduced by 22 times from a simple convolution operation.

2.1.2 Neuronal Bias and Activation

All the possible values which the parameters can take forms the solution space. As in the network shown in Figure 2.2(4), the possible solutions lies only on the line along the origin. Bias and Activation are required to increase the solution space beyond the origin and lying on just a line.

(i) Bias As shown in Figure 2.2(4), each hidden neuron takes the input and multiply it with the weight. For example, the hidden neuron 'h_2' takes the value $x_2 * w_2 + x_4 * w_4$. In a 2-D solution space where w_2 and w_4 are the parameters, the possible solutions lie only on the lines along the origin. This problem extends to higher-dimensional spaces as well. Therefore, a Bias value is added after multiplication of input with weight making $h_4 = x_2 * w_2 + x_4 * w_4 + b_4$ to increase the solution space. The solution (line) can now translate from the origin to other points along both the x and y axes. Bias is necessary to perform the affine transformation in the solution space. For a classification network, after completion of successful training, those weights and biases are learnt such that the network can extract discriminating features from the input data. This logic applies to FCL: one bias value for every neuron and CL: one bias value for every filter. Even after adding bias to FCL and CL, the function will still represent only a linear transformation. Hence the addition of multiple FCLs and CLs will not make much difference because they can all be collapsed into one linear learnable function.

(ii) Activation Layer The function that maps the real-world images to their class labels in case of a classification task can be highly complex and non-linear. A linear model may not be sufficient to learn complex features from the image. The activation layer is non-linear and is placed between the consecutive CLs or FCLs to improve the learning capability to account for such non-linear mappings. All neurons in the input layer may not

have useful information, so an activation function decides which neurons from the input layer must be fired to the output. An activation function also scales the input values between certain ranges without which the magnitude of neurons can explode or disappear because of the repeated multiplication. Non-linear activation functions allow the model to create a complex mapping between the network's input and output, essential for learning and modelling complex data, such as images, video, audio. Some activation functions, their properties, advantages and disadvantages are discussed below:

Sigmoid	**TanH**	**ReLU**	**LeakyReLU**	**Softmax**
$\frac{1}{1+e^{-x}}$	$\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$	$max(0,x)$	$max(0.1*x,x)$	$\frac{e^{x_i}}{\sum_{i=1}^{N} e^{x_i}}$

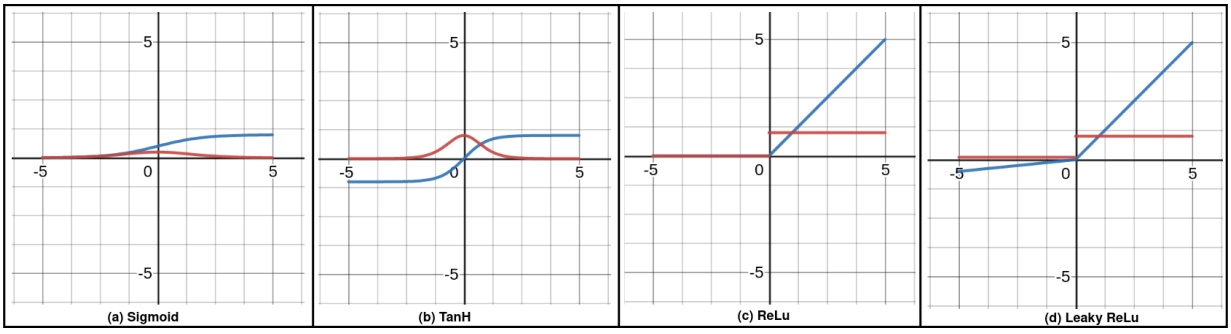


Figure 2.7: Graphs of activation functions (in Blue) and their gradients (in Red).
(a) Sigmoid- Range : (0, 1), Center : 0.5;
(b) TanH- Range : (-1, 1), Center : 0;
(c) ReLU- Range : $(0, \infty), Center : 0$;
$(d) LeakyReLu - Range$:$(-\infty, +\infty)$, Center : 0

- **Sigmoid:** It is an activation function which is smooth and avoids impulsive changes in the output values and therefore its continuously differentiable. Output values are bounded between 0 and 1 centered at 0.5 as shown in (a) of Figure 2.7. It can also be seen that the gradient values are significant only for values between -3 and 3. This makes the gradients insignificantly small and saturates for values beyond -3 and 3. When the gradient is 0, network does not learn anything and this phenomenon is called the Vanishing gradient problem. Also since the outputs are not centered

around 0, negative and positive values in the input cannot be defined clearly and output of all the neurons will be positive.

- **TanH:** This activation function is very similar to Sigmoid function which is scaled so as to be centered around 0. Output values are bounded between -1 and 1. Hence, the input to the next layer after TanH activation will not always carry the same sign. The gradient is steeper than Sigmoid's gradient, as shown in Figure 2.7, but the vanishing gradient problem still exists.

- **ReLU (Rectified Linear Unit):** It sets the output values to 0 when input values to the function are 0 or less than 0. This means neurons will be deactivated when the linear transformation from the previous layer results in a negative value. Since only fewer neurons are activated, ReLU is computationally very efficient. However, as shown in Figure 2.7(c), its gradient does not exist for the negative side of the graph. So the weights and biases for certain neurons will not get updated during the backpropagation process, and this phenomenon is called the Dying Neuron problem.

- **Leaky ReLU:** This is an improved version of ReLU and is defined to address the Dying Neuron problem. It fixes the problem by giving a tiny bit of positive slope for negative values. Hence the gradient of the left side of the graph comes out to be a non zero value as shown in (d) part of Figure 2.7.

- **Softmax** This can be seen as a combination of multiple sigmoids. Since the sigmoid function scales the value between 0 and 1, it can be seen as the probability of that data point belonging to one of the two classes. Therefore, softmax can be used for multi-class classification. It is usually used in the last layer because it scales the values between 0 and 1 such that they sum upto 1. Thus can be used as the probability of the input value belonging to a class.

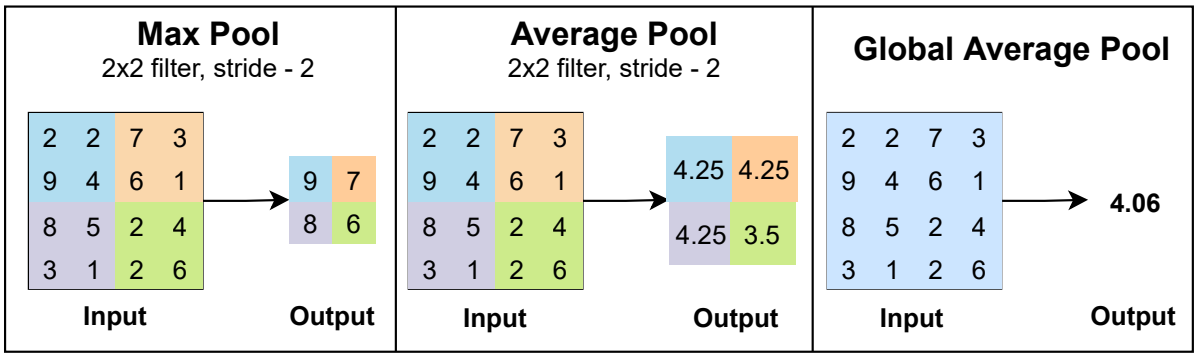


Figure 2.8: Different types of pooling layers. Max Pool extracts the maximum value in the patch. Average Pool computes and outputs the mean of the patch. GAP outputs a single value for the entire input.

2.1.3 Pooling

The pooling layer follows the convolution layer to extract the vital information from the output of the conv layer. The pooling layer, as shown in Figure 2.8 operates upon each feature map depth-wise separately to create a new set of the same number of pooled feature maps. For practical reasons, it is almost always 2×2 pixels applied with a stride of 2 pixels in both directions. The pooling layer will always reduce the size of each feature map by a factor of 2, e.g. each dimension is halved, reducing the number of pixels or values in each feature map to one quarter the size. In any image, the locality property holds, stating that neighbouring pixel intensity values will be similar. For example, in the case of the input cat image in Figure 2.3, most pixels of a feature map could be describing the stripe pattern of the cat skin. Pooling eliminates some redundant information making the features invariant to local variations and the cumulative pooling operations across layers enables extraction of high level (more global) featurs. Following are the popular types of pooling.

- **Average Pooling:** Itcalculates the average value for each patch on the feature map.

- **Maximum Pooling (or Max Pooling):** It calculates the maximum value for

each patch of the feature map.

- **Global Average Pooling (GAP):** It down samples the feature by averaging the entire feature map into a single value instead of acting on patches like other pooling layers discussed earlier.

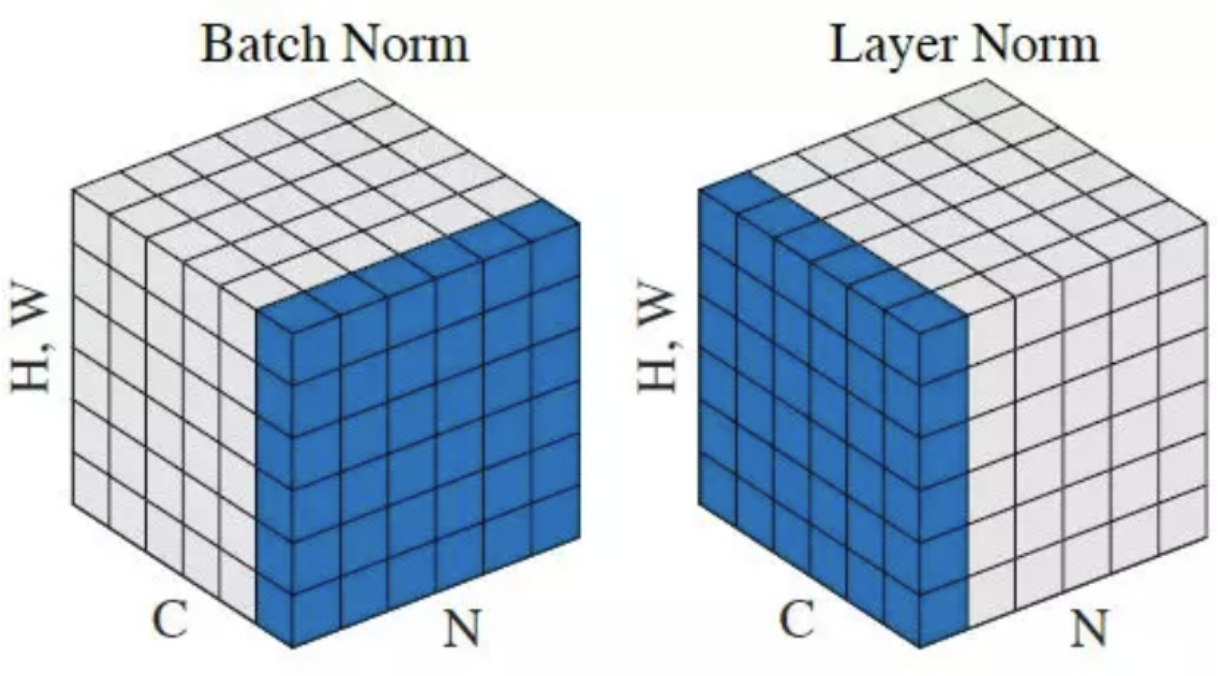


Figure 2.9: Different types of normalization. Batch Normalization works across the samples of a batch while Layer Normalization works across the channels/features. H: Height and W: Width of the image, C: Channels, N: Batch Size.

2.1.4 Batching and Normalization Layer

The first layer of a neural network that receives the input data for processing is the input layer. During training, the training samples are shown multiple times to the network. *Batching* refers to packing a certain number of training samples to show the network at each time. *Normalization* is the method in which input to every layer is processed for efficient training.

(i) Batching: Showing the full training set once to the network is called an epoch. Training data can be shown in batches of different sizes to the network. N in Figure 2.9 refers to the batch size or the number of samples in a single batch. Each epoch can have multiple iterations based on the batch size. Typically there are three ways of batching, as shown below.

- All the training samples (say Z) can be made into a single batch, and the number of iterations in each epoch will be equal to 1.
- Each training sample is made into a batch, making batch size equal to 1 and number of iterations equal to Z.
- There is a midway where a few training samples say (m) form a batch. Then, the batch size equals m, and the number of iterations equals Z/m.

At the test phase, either the whole test set or mini-batches are passed once to the trained network.

(ii) Normalization: While input to the first layer is usually normalized, inputs to the subsequent layers also require Normalization during training. Since the input to a layer depends on the parameters of the previous layer, and parameters are updated in every iteration during training. The change of parameters causes a change of distribution of the generated features, which are fed to the next layer. This phenomenon un-stabilizes the network performance, and Ioffe Szegedy's in [23] has defined it as "Internal Covariate Shift". Features passed through the multiple layers (conv, pool and activation layers) of the deep neural network will suffer distortion when it reaches the last layer. Normalizing the inputs of each internal layer improves the performance. Following two methods of Normalization are frequently applied in practice:

- **Batch normalization** (BN) [23]: This technique standardizes the input for each mini-batch of size N by keeping the data scaled (mean 0, and standard deviation 1)

during the training, facilitating faster convergence. The input features are normalized across the batch as shown in Figure 2.9(a). The average of mean and standard deviation over all the batches is used while testing. The data of every mini batch x_i, is subtracted by its mean μ_β and divided by its standard deviation σ_β to generate the normalized data ($\hat{x_i}$). In the following, input values are represented as x_i over a mini-batch $\beta = x_{1...N}$ and output value is represented as $y_i = BN(x_i)$.

$$\mu_\beta = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad \sigma_\beta^2 = \frac{1}{N}\sum_{i=1}^{N} x_i - {\mu_\beta}^2 \qquad \hat{x_i} = \frac{x_i - \mu_\beta}{\sqrt{{\sigma_\beta}^2 + \epsilon}} \tag{2.1.1}$$

- **Layer Normalization**: Batch normalization works fine for CNN models. However, it fails for recurrent networks, designed to handle sequential data where a batch contains the input of consecutive time steps. Hence, each sample from each time step has related information but are not similar information. Therefore, shifting and scaling along all the samples in a batch may not work well. Also, batch normalization of every sample is computation and space inefficient as mean and standard deviation from every step need to be stored. For sequential data, every sample in the batch has to be normalized independently, called layer normalization (LN). This method normalizes data across the channels/ feature maps instead of batch samples, as shown in Figure 2.9 (b). Implementation issues with constraints on GPU memory is often a problem forcing to keep the batch size equal to 1. LN does not impose size constraints on the batch because LN is implemented across all features of a single sample irrespective of the batch size.

2.1.5 Training

A simple neural network with two inputs, one hidden layer with two neurons (h_1, h_2), and one output layer having two output neurons with corresponding weights and biases is

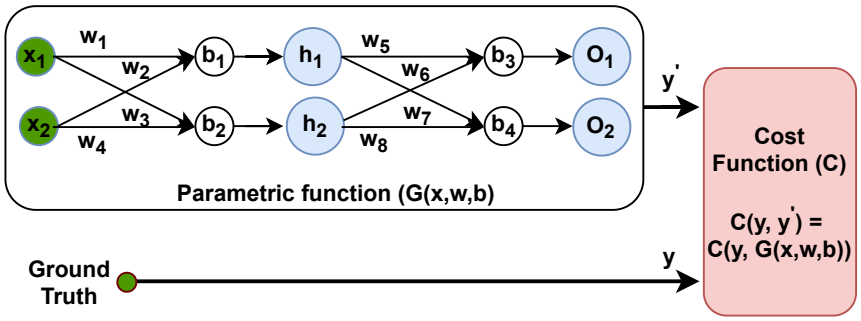


Figure 2.10: Simple neural network with Parametric function (G) and Cost function also called the Objective function (C)

shown in Figure 2.10. Since every node is connected to every other node in the next layer, this is a fully connected neural network. This model approximates some complex non-linear parametric function (G) with Inputs, Weights and Biases as the parameters. Each input is passed forward through (G), and it produces an output ($\hat{y}$). The Cost/Loss function (C) compares the output ($\hat{y}$) predicted by G and the ground truth (y) extracting all aspects of the model down via a scalar value called as **error**. The error is backpropagated through G, and an optimization algorithm is applied to every learnable parameter to find the optimal values such that the error is minimized in the next step.

(i) Cost/Loss Functions: Aim of the training procedure is to optimize the parameters to reduce the error gradually. However, choosing a suitable loss function is a challenging problem. Depending on the problem to solve, the following are some cost functions applied in practice:

- Mean Squared Error (**MSE**) is popular for function approximation (regression/estimation) problems, where the correlation between actual and predicted values is found, as described in Chapter 3. MSE is used for continuous target value, and its goal is to maximize the likelihood under the assumption that the data is normally distributed.

- The cross-entropy (**CE**) error function outputs a value between 0 and 1. So, it is

often used for classification problems, where outputs are interpreted as probabilities of membership to a class. It is used together with the sigmoid or softmax non-linearity in the last layer of the network. The goal of CE is to maximize the likelihood of predicting the true class of the input data.

- **Log Sum Exponential Pairwise Loss** (LSEP) function [24] or ranking loss, finds the relative distance between the labels; unlike MSE and CE, which are used to learn to predict the label directly. This cost function is expressed as,

$$Loss_{LSEP} = \log(1 + \sum_{v \in Y_i} \sum_{u \in Y_i} \exp(f_v(x_i) - f_u(x_i)), \tag{2.1.2}$$

where f_X is the label prediction function that maps the document vector x into K-dimensional label space representing the confidence scores of each label (K equals to number of unique labels). $f_u(x_i)$ and $f_v(x_i)$ are the v and u -th element of confidence scores for the i-th instance in the dataset, respectively. Y_i is the corresponding label set for the i-th instance in the dataset. As this function is smooth and easily optimizable for multi label classification tasks, it is used in this work to classify Xray tags, as described in Chapter 4.

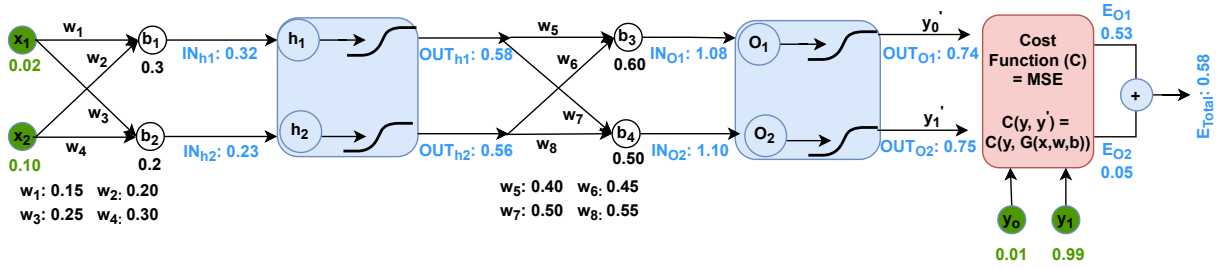


Figure 2.11: Input and Ground Truth nodes are shaded in Green, Parameter nodes that are trainable are represented in Black, Neurons which are computational nodes are shaded in Blue and the computed values are represented in Blue. Computational nodes are in two parts - neuron followed by the Sigmoid activation function. Cost function used here is Mean Square Error (MSE)

(ii) Forward pass: Figure 2.11 shows the forward pass on an example toy network **G**.

Incoming values to the computational nodes - h_1, h_2, o_1 and o_2 are linear combination of inputs, weights and biases. For example $IN_{h1} = w_1 * x_1 + w_2 * x_2 + b_1$ and $OUT_{h1} = \frac{1}{1+e^{-IN_{h1}}}$. Following the same procedure through till the end, network predicts two outputs $y_0^{'}$ and $y_1^{'}$. The cost function evaluates the error E_0 and E_1 between predicted outputs and ground truth y_0 and y_1. E_{Total} is the sum of both the errors. We want to update the weights so that the predicted output is closer to actual output, thereby minimizing the error. We want to know how much change in each weight will effect the total error E_{Total}. This is learnt using a process called **Backpropagation**.

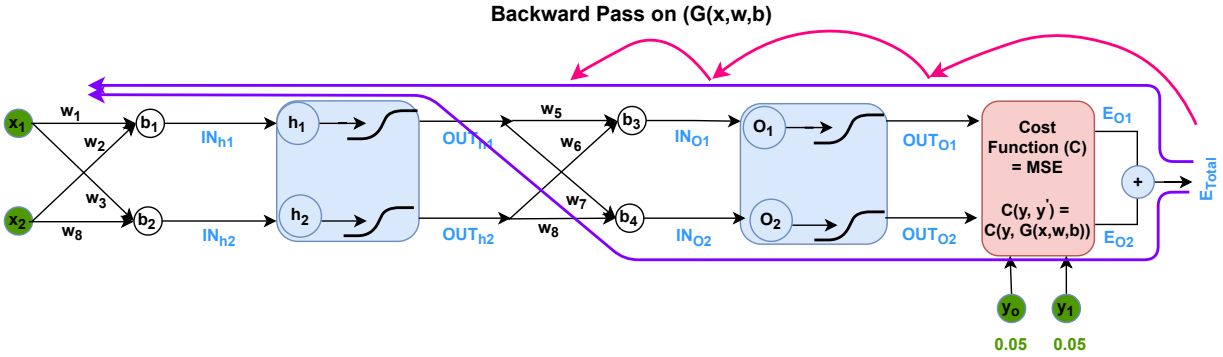


Figure 2.12: Pink arrow denotes the path of required gradients to update w_5. w_1 effects the E_{Total} along two paths so the two purple lines denote the trace from E_{Total} back to w_1.

(iii) Backpropogation: As shown in Figure 2.12 Backpropagation algorithm is applied during the backward pass on **G**. This algorithm traces back from the model's output through the different neurons involved in generating that output, back to the original weight applied to each neuron. For example, to know how much change in w_5 affects the total error find $\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial OUT_{o1}} * \frac{\partial OUT_{o1}}{\partial IN_{o1}} * \frac{\partial IN_{o1}}{\partial w_5}$. All these gradients are computed using the Chain Rule. All the individual gradients on the right-hand side of the above equations mentioned here can be computed directly since the numerators of the gradients are explicit functions of the denominators. $\frac{\partial E_{total}}{\partial OUT_{o1}}$ is the incoming gradient which flows in the backward pass and gets multiplied with the rest of the terms which are the local gradients ($\frac{\partial OUT_{o1}}{\partial IN_{o1}}$, $\frac{\partial IN_{o1}}{\partial w_5}$). Local gradients do not need information from the future to get

computed, so they get taped during the forward pass itself. In another case, OUT_{o1} affects both OUT_{o1} and OUT_{o2}. If there is more than one path to a variable from E_{total} as in case of w_1 then, we multiply the edges along each path and then add them together as shown by the purple lines of Figure 2.12. We repeat this process to all the weights and biases.

(iv) Optimization: The goal is to come up with such weights and biases (parameters of the model), so that accurate prediction can be made for the given task. In the above simple toy network, there are 12 trainable parameters (8 weights and 4 biases). Practical neural networks can typically have millions of parameters. All the weights and biases in the model are unknown - that is, too many parameters for which one needs to formulate an optimization problem. The **Optimization Algorithm** navigates in the parameter space to find an optimal solution in order to make better predictions in each iteration. The constraint to this optimization problem is to minimize the objective function **C**. C calculates the error of the model. To decrease the error value, the gradients are subtracted from the current weights. Its optionally multiplied with a learning rate η. For example, w_5 is updated with $w_5^+ = w_5 - \eta * \frac{\partial E_{total}}{\partial w_5}$. This is given by the optimization algorithm called **Gradient Descent (GD)**. It suggests an optimal weight for each neuron which results in the most accurate prediction. Deep learning is full of gradient-based methods. The gradient descent is an algorithm to find the minimum of the objective function, which has to be differentiable and continuous. GD can be applied per sample or on all the samples. Per sample leads to a noisy trajectory during optimization due to oscillations. In practice, it is applied in mini-batches to enable parallelization due to constraints imposed by the current GPUs. Many of GDs variants like **Adam/ RMSProp/ Adagrad** are also used alternatively. Finally, after updating all the weights, the inputs are fed again through the network to find that the error decreases. After repeating this process several times, the error stagnates.

In the following sections, we will discuss some aspects and deep learning architectures that are relevant to the work done in this thesis.

2.2 Autoencoder

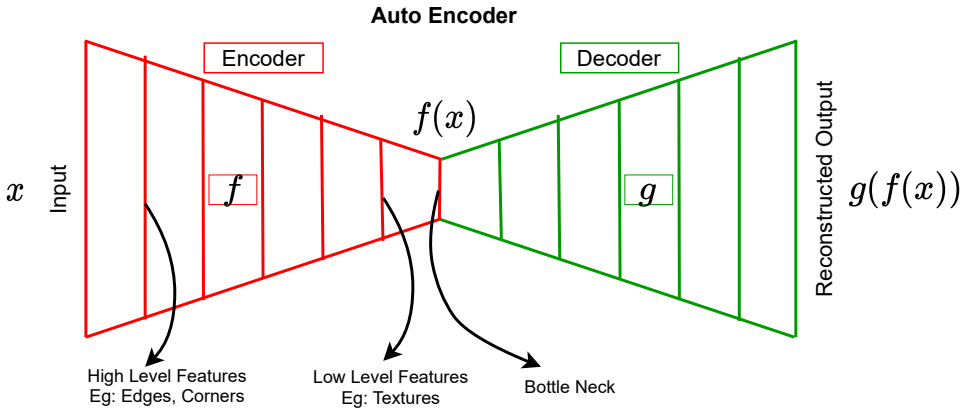


Figure 2.13: Initial features are basic low level where as towards bottle neck high level features are extracted.

The basic *Autoencoder* is a design trained to learn the copy of the input [25]. The network has two parts encoder that encodes the input $h = f(x)$ and decoder that produces a reconstruction $x = g(f)$. There is not much practical value to construct the input using $x = g(f(x))$. Actually, autoencoders as in Figure 2.13 are trained to extract the condensed representation of the information that can be suitably utilized to reconstruct the input. This representation is generally much smaller in size than the original input as it selectively eliminates the redundancies. Here h is a compressed representation seen as the bottleneck imposed by the Encoder to learn compression of the original input. General representation of autoencoder is $r = g(f(x))$ where h is the internal latent representation of input data. It can also be used for dimensionality reduction so as to represent the data compactly. Training must ensure that autoencoder captures the input features effectively and also not memorize the input, which causes overfitting. Usually, to manage this trade-off, the objective function has a Loss term and a Regularizer term. A bottleneck can be enforced

by reducing the number of neurons in the hidden layers of an FCN architecture as shown in Figure 2.4(1). We want the network to learn latent features.

Convolutional Autoencoders utilizes 2D/3D convolutional filters and can reduce the input into a latent representation and reconstruct the input from it. It can also be seen as a function approximator for one-to-one mapping functions in a supervised learning paradigm. This implies that if each input sample and output sample have correspondence to each other, like in the case of T1 to T2 weighted image as in Chapter 3, the Encoder-Decoder design is expected to learn the mapping. Each convolutional layer in Encoder from start to bottleneck learns low levels features like edges and corners to high levels features like texture and shape. This inturn reduces the contextual spatial information by reducing the size at the bottleneck. Hence, the decoder, in its upsampling process needs to learn the spatial relationships well.

2.3 Metric Learning

Distance between any two data points can defined as a metric and it has to be suitably designed utilising the priory knowledge about the data domain. Metrics must be designed to suit the data and the task at hand. Automatically learning such a metric is called Distance Metric Learning [26] or simply Metric Learning. A data point can be transformed and projected into a new point in a different space, and that transformation can be learned, such that the distance in this new space is better suited for the task at hand (for example, better separability between the classes for a classification task.). The transformed point is called as the embedding. Especially when the number of classes is very large or when the data is skewed/imbalanced, i.e there are very few samples in a class compared to the others, metric learning can be quite useful to learn discriminating features. In Chapter 4,

we have a classic example of data imbalance where the amount of abnormal Xray images is far fewer than normal Xray images for binary classification.

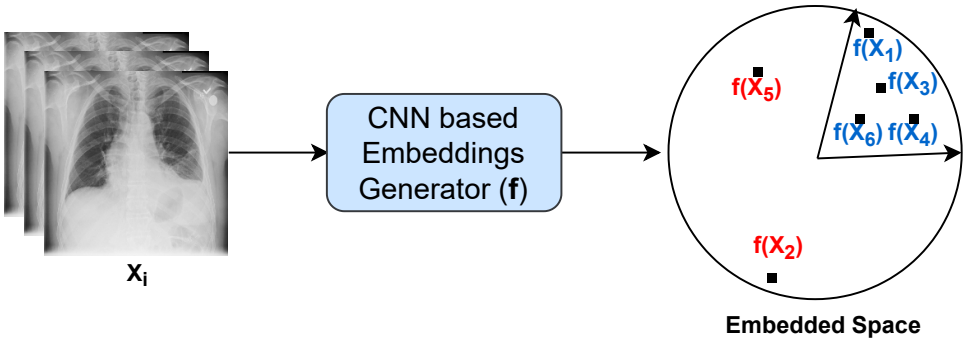


Figure 2.14: CNN based Embeddings Generator with $X_i \in R^d$ is input and feature embedding vector $f(X_i) \in R^m$ as output while $m << n$. $f(X_2), f(X_5)$ in red are negative samples while $f(X_1), f(X_3), f(X_4), f(X_6)$ in blue are positive samples clustered together.

The Deep Metric Learning proposes a CNN based Embeddings Generator (EG) as shown in 2.14. It will project normal Xray images of healthy patients (for example: X_1, X_3, X_4, X_6) as close as possible to each other within a cluster while throwing the abnormal Xray images of unhealthy patients (X_2, X_5) far away from the normal cluster in the embedded space by learning a suitable transformation (f). After extracting these embeddings, a classifier is trained to perform the binary classification. Embeddings will be learned based on the criteria of the task. For a given anchor image(a), a positive image(p) is similar to the anchor, while a negative image(n) is different from the anchor. Similarity constraint can either be

- Absolute: $D(a,p) = 0 \;\; \& \;\; D(a,n) > \alpha$ (or)
- Relative: $[D(a,p) - D(a,n) + \alpha] > 0$

where $D(a,p)$ is the distance between anchor and positive, D(a, n) is the distance between anchor and negative. D is some distance measure (say Euclidean) between the learned representation of input images. Contrastive loss uses the absolute similarity constraint and enforces $D(a,p)$ to be close to 0 and $D(a,n)$ to be greater than a constant α. Relative

loss enforces $D(a, n)$ to greater by atleast an α over $D(a, p$. Triplet loss uses relative similarity constraint and trains the Embeddings Generator (EG) network in Chapter 4. The networks which use the above kind of losses to differentiate between images but not classify them are called **Siamese Networks**. These embeddings can also be learned so that the number of classes in the dataset does not matter. These features can be general enough to classify samples of unseen classes too.

2.3.1 Triplet loss and Hard Negative Mining

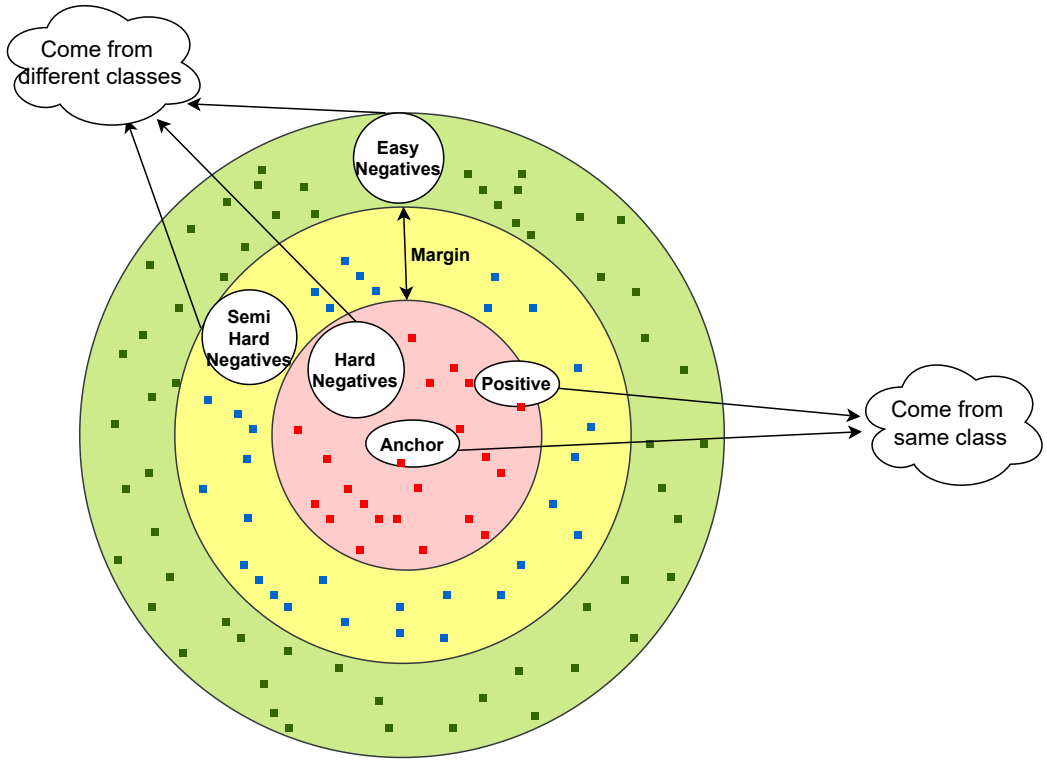


Figure 2.15: Placement of Easy, Semi Hard and Hard negatives on the 2D embedded space. With anchor and positive sample in ovals, the squared points indicate all possible places where negative samples can lie in the Embedded space. Distance between two points signifies the closeness between the points.

Contrastive loss is one sided which means at any time, network can learn to either bring anchor and positive closer or push away anchor and negative farther from each other.

Triplet loss solves this problem by training for both goals simultaneously. Image triplets must be built out of the data to compute Triplet loss: <anchor, positive, negative>. In Chapter 4, there are two classes of data - abnormal patients data and normal patients data. Images from the same class are considered as similar and dissimilar otherwise. The loss function is given as

$$L(a, p, n) = max(0, D(a, p) - D(a, n) + margin) \tag{2.3.3}$$

As shown in Figure 2.15, minimizing the above loss ensures to maintain a margin (α) between the D(a,p) and D(a, n). Thus, it ensures that the positive samples are all collapsed into a compact cluster, whereas the negative samples are away by at least a margin from the positive cluster. This model does not care about intraclass variability, which is very good for the binary classification problem in Chapter 4, where images has to be classified in a hierarchy - first between normal and abnormal and then within the abnormal images.

Triplets are generated either in an off-line mode where we manually generate the data and fit into the network or online mode where we feed a batch of training data, and triplets can be randomly created during the run time. Both techniques increase the chances of finding hard triplets and train the network faster. Easy triplets can be a tuple of (cat-1, cat-2, dog) and hard triplets can be a tuple of (cat-1, cat-2, tiger). Cat and tiger form hard negative while cat and dog form easy negative. Its easier for the network to differentiate and learn from easy triplets than the hard triplets. Therefore, hard triplets give higher loss to the network enabling better training than 0 loss given by easy triplets. Semi-hard triplets and semi-hard negatives exist between these two extremes. Hard negatives are mined for training efficiency. Either all valid triplets are batched but loss is averaged only on hard and semi-hard triplets or for each anchor select only those triplets for whom $D(a, n) < D(a, p)$ (hard negatives) and $D(a, n) < D(a, p)+margin$ & $D(a, n) >= D(a, p)$ (semi-hard negatives) is true and batch them for training.

2.4 Transformers

Many neural network architectures are developed to solve a task that involves transforming a sequence of input to an output sequence, also called sequence transduction [27]. Most such architectures are recurrent neural networks (RNNs). The disadvantage with recurrent neural networks is the lack of parallelization and learning from longer distances. This can especially affect while having memory constraints with batching.An architecture proposed in [27] called **Transformer** solely uses attention mechanisms instead of recurrence or convolutional layers to solve seq-to-seq modelling and can be trained efficiently.

2.4.1 Attention

Transformers use *Attention* mechanism to process one specific aspect of a complex input and continues through the input sequence until completion. It tries to compute the importance of all the tokens with respect to a given token (t). This helps to reconstruct that token (t) back by utilizing its relevant information from all other tokens. Transformers utilizes two types of attentions: Self attention (intra domain) and Cross attention (inter domain).

2.4.1.1 Self and Cross Attention

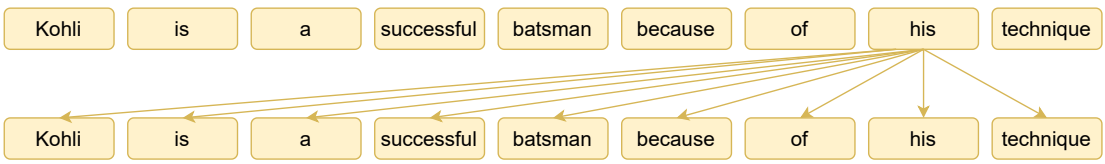


Figure 2.16: Self Attention functionality with a high level demonstration

Self-attention is a concept introduced in [27] and is referred as attention between the different positions of the same sequence. It computes the relevance of every symbol of the

sequence to all other symbols of the same sequence. For example, as shown in Figure 2.16 in the English sentence "Kohli is a successful batsman because of his technique". Here, "his" refers to Kohli. Hence, self-attention on this sentence outputs a higher relevance score to symbol "Kohli" with "his" or "batsman" over other symbols like "of" and "technique". Cross attention is computed between two different sequences and is typically used in decoder because here the output sequence needs to be generated using latent representation of the input sequence. In chapter 4, we used two encoders to encode information from two different data types : Images and Text. Therefore Decoder has two Cross Attention layers to compute attention across the outputs of both encoders.

2.4.1.2 Scaled Dot-Product Attention

Attention in transformer is computed using scaled dot product as shown in Algorithm 1. Scaled Dot-Product attention computes Query ($\mathbf{Q}$), Key ($\mathbf{K}$) and Value ($\mathbf{V}$) from the input embeddings ($I = (I_1, I_2, ..., I_t)$). These are generated by multiplying the each vector of input embedding (I) with three matrices - $\mathbf{W}_q$, $\mathbf{W}_k$ and $\mathbf{W}_v$ respectively, which are learned during the training process. Such transformation enables that each position is efficiently attended to all other positions as explained below.

Scaled Dot-product utilises Query, Key, and Value to calculate a relevance score for each symbol (I_s) in the input sequence against every other symbol ($I_i, \forall i \in t$). The score will objectively determine how much each pair of symbols are co-related. The relevance score is obtained by performing the dot product of every $\mathbf{Q}_s$ with $\mathbf{K}_i \forall i \in t$, providing us with a t-dim score for each symbol $\mathbf{V}_s$. This vector is normalized by $\sqrt{d}$, where each symbol in Query (Q_i) is d-dimensional, of the output to have lower variance. We perform softmax function across the attention scores, which converts them to corresponding attention probabilities ($\mathbf{p}$). Finally, it attends to each $\mathbf{V}$ wrt the corresponding attention

probabilities by multiplying $\mathbf{V}_s$ with $\mathbf{p}$.

Although the above explanation was done through single embedding vectors, implementation is done by packing all these embeddings into a matrix. All the positions will be filled in O(1) time. $\mathbf{z} = softmax(\frac{\mathbf{QK}^T}{\sqrt{d}})\mathbf{V}$. We can see that $\mathbf{Q}$ and $\mathbf{K}$ must be of the same dimension d while $\mathbf{V}$ can be of a different dimension.

Algorithm 1: Scaled Dot-Product Attention

Input: Input sequence $I = (I_1, I_2, ..., I_t)$
Output: Latent representation (z)
1 $\mathbf{Q} = I * \mathbf{W}_q$, $\mathbf{K} = I * \mathbf{W}_k$, $\mathbf{V} = I * \mathbf{W}_v$
2 Attention Matrix $(\mathbf{A}) = \mathbf{Q} \cdot \mathbf{K}^T$
3 Normalized Attention $(\mathbf{A}) = \frac{\mathbf{A}}{\sqrt{d}}$
4 Attention probabilities $(\mathbf{p}) = softmax(\mathbf{A})$
5 $\mathbf{z} = \mathbf{p} * \mathbf{V}$ ▷ Latent Representation

2.4.1.3 Multi Head Attention

The attention of a word in a sentence/text depends upon the context. Since the context may vary and have short/long term dependence, learning multi contextual attention is beneficial to incorporate complex language semantic structure. This can be done by learning multiple $\mathbf{W}$s, so as to choose the best/most compatible fusion between different sets of relevance values obtained. As seen in the example Figure 2.16 symbol "his" has higher relevance with "Kohli", "successful" and "batsman". As explained in Figure 2.17, one set of $\mathbf{W}$s may not be able to learn every relationship because each of these pairs is related to each other differently - identity, adjective, name, role etc. Output from all the heads $\mathbf{z}_0$, $\mathbf{z}_1$, $\mathbf{z}_2$,, etc,. are all concatenated before putting through the feed-forward layer. Another large $\mathbf{W}$ needs to be learnt during the training to get the final single output vector $\mathbf{Z}$.

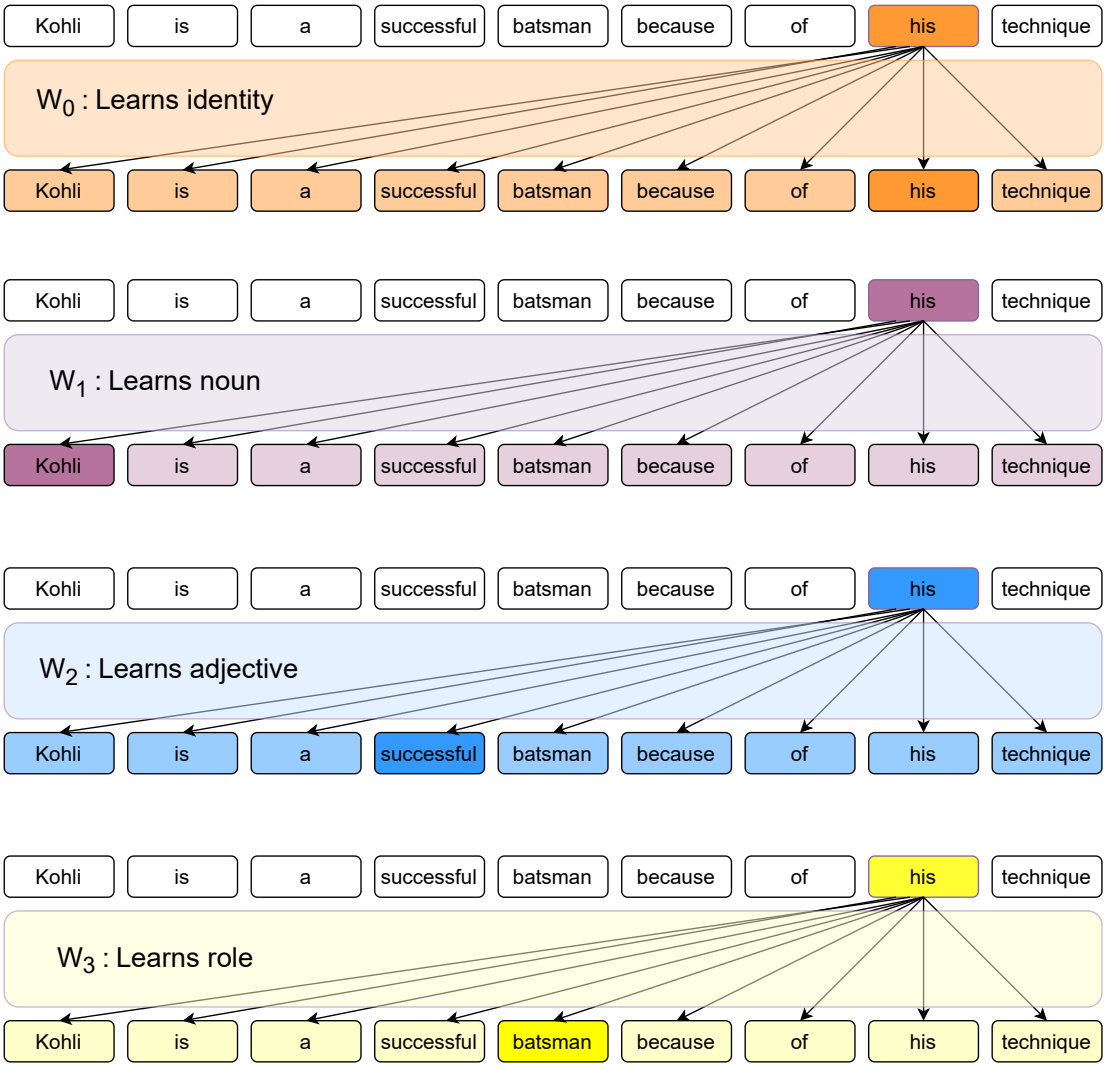


Figure 2.17: Usefulness of Multi Head Attention.

2.4.2 Encoder

Transformer is an Encoder-Decoder based model. Encoder converts the input's symbol representation ($\mathbf{x}$) to a continuous representation ($\mathbf{z}$) in a much more compressed form. Encoder contains a stack of identical layers, each composed of two sub-layers: Self Attention layer and Fully Connected layer. In order to obtain a robust and multi contextual self attention, multi-head attention has been proposed as explained below. Finally the output has been augmented with residual connection and normalized using standard layer

normalization as given below:

Output of each layer = *LayerNormalization(ResidualConnection(*$\mathbf{x}$, *sublayer*($\mathbf{x}$)).

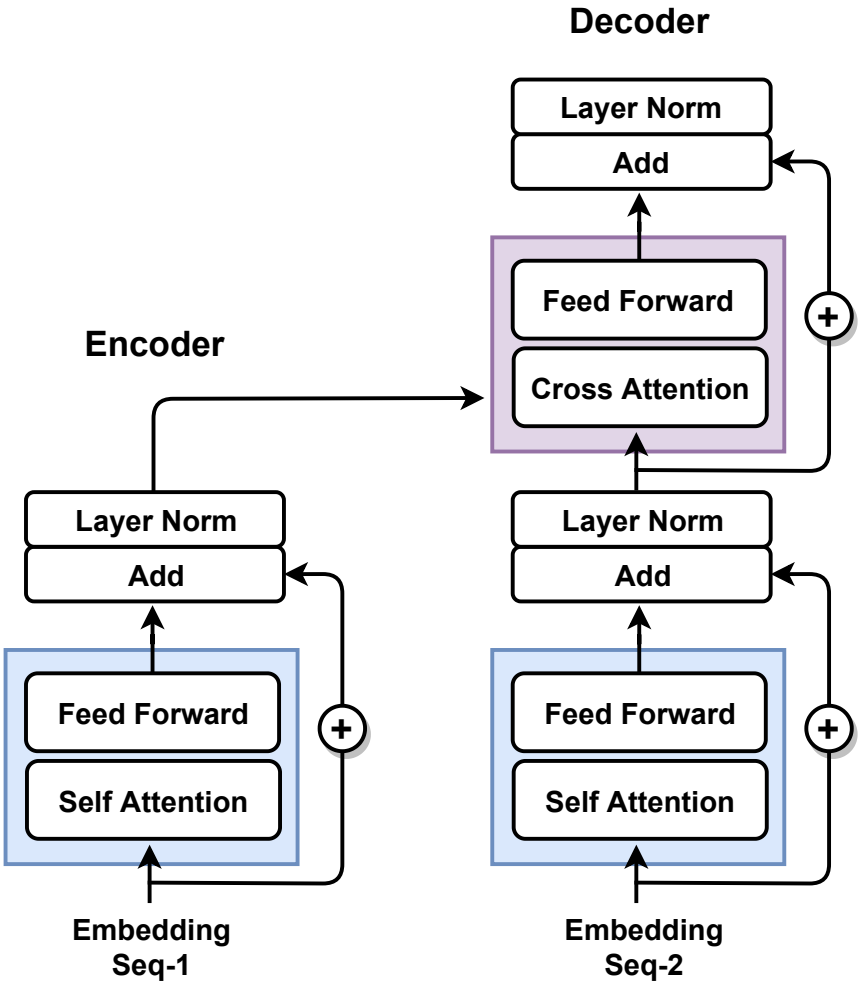


Figure 2.18: Encoder-Decoder architecture of a Transformer with single layer. Output of the encoder is used by the decoder to compute Cross Attention.

2.4.2.1 Embedding

Any network can only work on vectors; therefore, a word needs to be first converted into a vector using word-to-vector embeddings. Word embedding happens in the first layer of both the Encoder and Decoder for the language translation task. However, in this work,

since encoders process the images and tags, word embeddings are used only in the first layer of the Decoder. The size/dimension of the input vectors would be the length of the longest sentence in the training data. These vectors pass through the sub-layers (Self-Attention and Feed Forward) as shown in Figure 2.18.

2.4.2.2 Positional Encoding

All the operations in transformers are realised using matrix multiplication and hence its output is positional invariant. In seq-to-seq translation, positional information plays a pivotal role but transformers do not capture that information. In order to address this issue, transformer adds a vector called position vector to the input embedding to identify the position of each word. Since the position vector needs to follow a pattern, it is possible to determine the distance between the symbols. There can be several candidates for positional encoding and following are some desirable properties to have [28]

- Unique encoding for every symbol in the sequence.
- Distance between any two symbols must be consistent for any sequence. Meaning has to be consistent across sentences.
- Should generalize to longer sequences. So values have to be bounded. Otherwise, longer sequences will have larger values. It also poses a problem when sequences are longer during testing than the ones in training.
- Must be deterministic.

As proposed in [27], transformers utilize a sinusoidal $\boldsymbol{d}$-dimensional vector which holds information about the position of a symbol in a sequence. It's each dimension can be obtained by following the formula:

$$\mathbf{PE}_{(pos,2i)} = sin(\frac{pos}{10000^{\frac{2i}{d}}})$$

$$\mathbf{PE}_{(pos,2i+1)} = cos(\frac{pos}{10000^{\frac{2i}{d}}})$$

PE stands for Positional Encoding, pos is the sequential position, i denotes the dimensional position, and d is the dimension of the positional encoding and it must be equal to the size of the embedding so that they can be summed up as proposed in [27].

2.4.2.3 Architecture

Algorithm 2: Encoder Architecture

Input: Input sequence ($\mathbf{x}_1$)
Output: Latent representation ($\mathbf{z}$)

1 $\mathbf{p} = \mathbf{x}_1 + PE$ ▷ Input is added with its positional encoding
2 $\mathbf{q} = SA(\mathbf{p})$ ▷ Self Attention
3 $\mathbf{r} = FF(\mathbf{q})$ ▷ Feed Forward layer
4 $\mathbf{s} = Add(\mathbf{p}, \mathbf{r})$ ▷ Residual connection
5 $\mathbf{z} = Norm(\mathbf{s})$ ▷ Layer Normalization

For a given input sequence x, the order of operations applied on input in the Encoder are: add positional embedding (PE), apply self attention (SA) which is a form of multihead attention followed by a feed forward (FF) layer, set a residual connection (Add) by adding the original input x to the output of SA, and finally apply layer normalization to the outcome to form the latent representation z. The flow from input x to z is succinctly given in Algorithm 2:

2.4.3 Decoder

Decoder also applies attention to the input while generating the output. For a given latent representation $\mathbf{z}$ from Encoder, Decoder generates a sequence of output $\mathbf{y}$, one symbol at a time in an auto-regressive fashion. Therefore, the self-attention (SA) layer in the Decoder is modified using a mask to prevent applying attention to future positions. ***Masking*** combined with the right-shifted output sequences during training ensures that

the prediction at position i is based only on the previously seen positions of the output less than i. Decoder has an additional cross attention (CA) layer between the two sub layers which works on the output of the Encoder relevant to the input sequence as shown in Figure 2.18. The order of operations on input sequence $\mathbf{x}_2$ with the latent representation z in Decoder is shown in Algorithm 3.

Algorithm 3: Decoder Architecture

Input: Input Sequence ($\mathbf{x}_2$), Latent representation ($\mathbf{z}$)
Output: Output sequence ($\mathbf{y}$)

1 $\mathbf{p} = \mathbf{x}_2 + PE$
2 $\mathbf{q} = Norm(Add(FF(SA(\mathbf{p})), \mathbf{p}))$ ▷ Self Attention
3 $\mathbf{r} = FF(CA(\mathbf{z}, \mathbf{q}))$ ▷ Cross Attention
4 $\mathbf{s} = Add(\mathbf{q}, \mathbf{r})$ ▷ Residual connection
5 $\mathbf{y} = Norm(\mathbf{s})$ ▷ Layer Normalization

2.4.4 Training Procedure

During training, Encoder looks at an input sequence and produces a set of key and value vectors **K** and **V** respectively. The Decoder uses them in the Cross-Attention layer, which produces an output of one position until a special end symbol is reached. The output of the Decoder at each step is fed to the Decoder to compute the next step. These are embedded with positional vectors, just like in the case of Encoder and fed to the Decoder. For example, if the dataset has 1000 unique vocabularies like in Chapter 4, the logits vector will be 1000 elements long, each element corresponding to the score of the unique word. Applying softmax on this will turn scores into probabilities, and the word with the highest probability will be chosen as the output at the given time step. While training, The loss function (cross-entropy) brings the probability distribution of the ground truth and the network prediction as close as possible.

2.5 CycleGAN

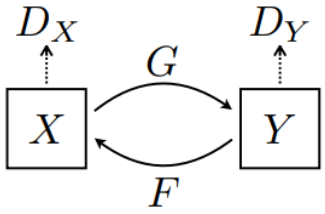


Figure 2.19: Block diagram of CycleGAN [1] training procedure.

Image to image translation means generating a synthetic version of an image with particular modifications, such as translating grey images to colour images. DL models typically require a large number of paired images while training to perform such a translation. Considering an image colorization task, it is practically very hard to get coloured version of an image for every corresponding grey image for training purposes. CycleGAN [1] is a technique involving the training of image-to-image translation without paired data. This form of training is called Unsupervised training, where source and target domain need not contain related information. As shown in Figure 2.19, the model uses two generative functions $G : X \rightarrow Y$ and $F : Y \rightarrow X$ with training samples $x_i \in X$ and $y_i \in Y$. The two functions G and F are trained together to produce images to confuse the corresponding discriminators D_Y and D_X respectively. The job of the discriminators is to identify if the images produced by generators is real or fake. Given a grey image x, G tries to produce $\hat{y}$ that is very similar to real y, good enough to fool D_Y. Similarly, D_X tries to discriminate between $\hat{x}$ produced by F and real x. The key idea is that generator and discriminator are made to compete to synthesise better images. Another example [2] of unpaired image to image transformation from domain of horses (A) to domain of zebras (B) based on the original work is shown in Figure 2.20.

The following losses are used to regularize the training procedure:

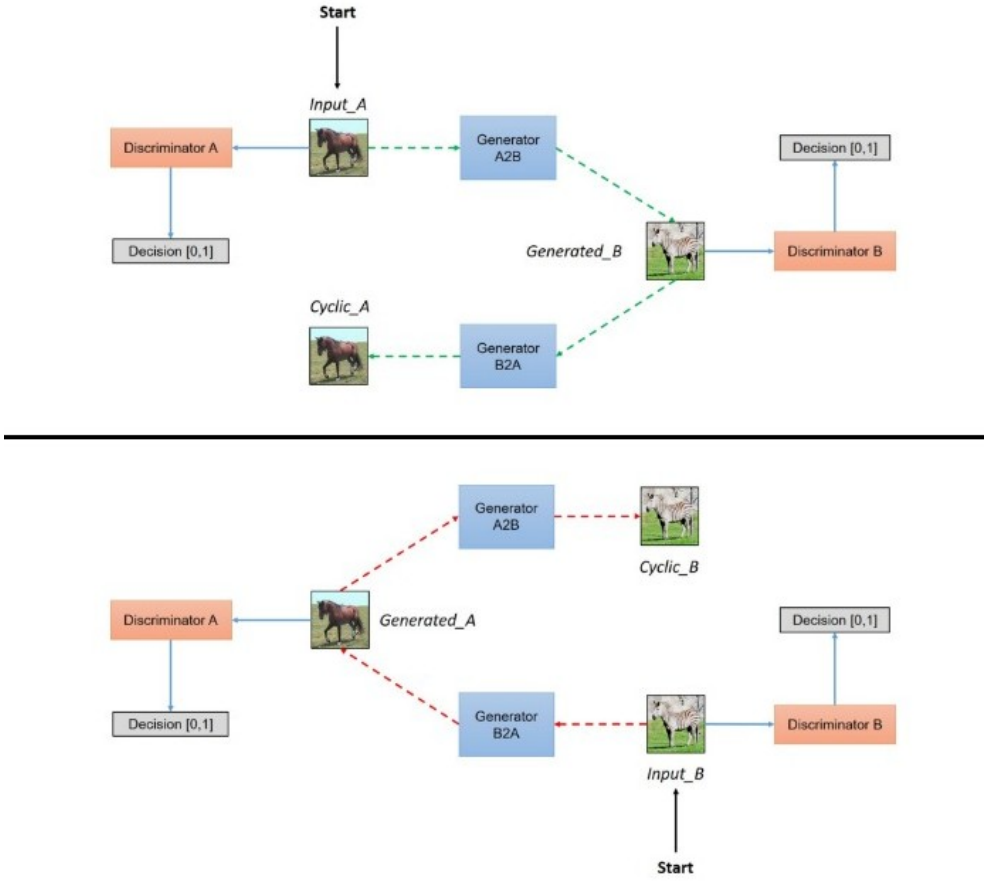


Figure 2.20: Image transformation from domain of horses(A) to domain of zebaras(B) from [2]

- Adversarial Loss: It is derived from the cross-entropy between real and generated distributions where $D_Y(y)$ is the discriminator's probability estimate that the real y is real, $G(x) = \hat{y}$ and $D_Y(G(x)$ is the discriminator's probability estimate that fake instance is real and **E** stands for expectation over all samples in the batch.

$$\min_G \max_{D_Y} L_{adv} = \mathbf{E_y}[logD_Y(y)] + \mathbf{E_x}[log(1 - D_Y(G(x))] \tag{2.5.4}$$

- Cyclic Loss: CycleGAN uses cyclic consistency loss to enable learning the unpaired mapping function. Cyclic Loss enforces G and F mappings to be reverse of each other and both mappings to be bijections. It encourages forward cyclic consistency:

$F(G(x)) \approx x$ and backward cyclic consistency: $G(F(y)) \approx x$ as is defined as:

$$L_{cyc}(G, F) = \mathbf{E_x}[||F(G(x)) - x||_1] + \mathbf{E_y}[||G(F(y)) - y||_1] \tag{2.5.5}$$

where $||.||_1$ denotes L_1 norm.

- Identity Loss: It says that, if an image $\mathbf{Y}$ is fed to the generator G, it should yield the real image $\mathbf{Y}$ or something closer to image $\mathbf{Y}$. The enforces to not modify the image because it already exists in the target class.

$$L_{idt}(G, F) = \mathbf{E_y}[||G(y) - y||_1] + \mathbf{E_x}[||F(x) - x||_1] \tag{2.5.6}$$

2.6 Conclusion

We have so far seen the motivations and contributions of this thesis. We also discussed preliminaries related to the techniques used in the later chapters. In the following chapters, we will see two prominent deep learning architectures viz Encoder-Decoder based models with convolutional layers for image transformation and attention-based Transformer architecture for image inference and transformation. Furthermore, we will see several submodels and methods to adapt these architectures to the task at hand and adhere to the data constraints.

Chapter 3

Medical Image Synthesis

This chapter covers an image synthesis task as a post-process of Medical Imaging. Particularly, we explore inter modality MR image reconstruction. Distinct but related information from multi-modality MR images such as T1 weighted, T2 weighted, FLAIR, proton density weighted (PDw), functional-MRI and diffusion-MRI provides several diagnostic benefits. Despite the importance of multi-modality MR images, sequential acquisition of multi-modality MR images, especially T2 weighted images with longer AQT, can be practically time-consuming, though beneficial for disease diagnosis. In order to enable the availability of T2 weighted images and shorten the acquisition time, a novel deep neural network based solution has been proposed to reconstruct T2 weighted images (T2WI) from T1 weighted images (T1WI).

The proposed network involves a Reconstruction Module (RM) augmenting in parallel to the Domain Adaptation Module (DAM), which is an Encoder-Decoder architecture with a Sharp Bottleneck Module (SBM) described in Section 3.2. The proposed network can reconstruct a 3D T2 weighted image's volume in approximately 42 and 46 seconds, without using any T2 information and with using T2 under-sampled k-space respectively. This can significantly reduce the total acquisition time. As a proof-of-concept, we demonstrate the

transformation with the proposed approach for normal brain MRI images with negligible qualitative artefacts and quantitative loss as described in Section 3.4. The testing has been done over two public datasets: (i) MICCAI challenge dataset [29] and (ii) HCP dataset [30], and the results are compared via standard performance parameters (such as PSNR, SSIM, MAE) suitably with the recent state-of-the-art [3]. The proposed network has shown significant improvement, and superior reconstruction as compared with a recent approach in [3].

3.1 MRI Inter-Modality T1 to T2 Reconstruction

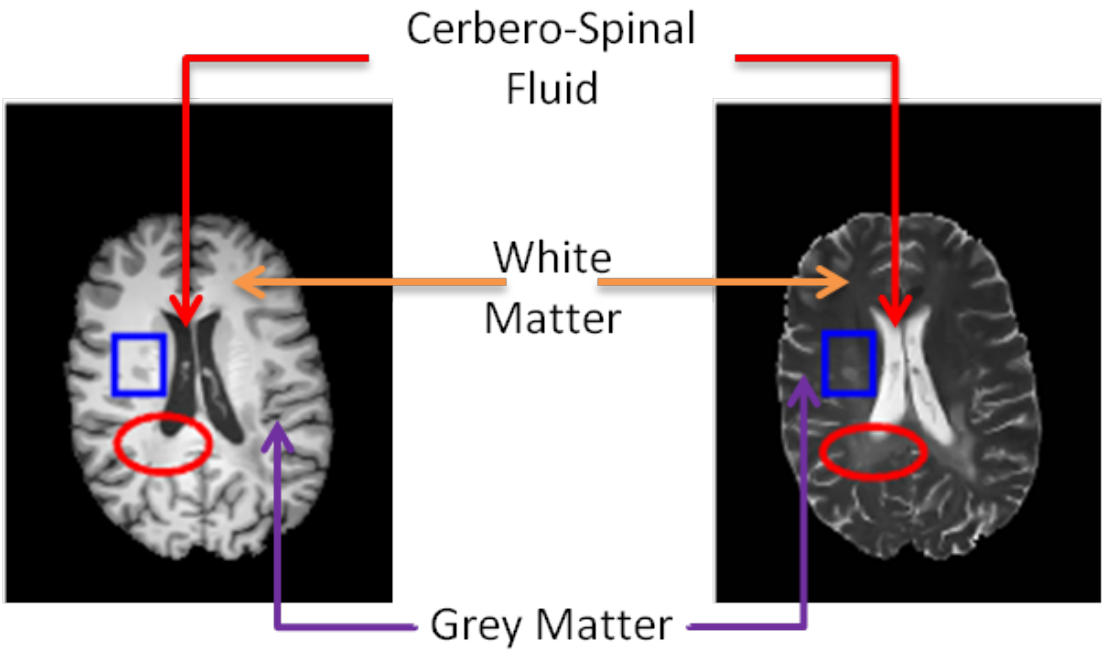


Figure 3.1: Illustration of T1 and T2 modality images (registered for the same subject). White matter appears bright in T1 but dark in T2. Depending upon the lesion's characteristics, it may behave similarly (in the blue box (left is hypo and right is hyper)) and differently (in the red ellipse, left is iso and right is hyper) in T1 and T2 weighted images. Iso-intense regions are hardly perceivable in any modality.

Hypo-intense information in T1WI becomes hyper-intense in T2WI and the contrast

in the colour of CSF (Cerebral Spinal Fluid) in T1WI and T2WI can be seen in Figure 3.1. The disease diagnosis can often be benefited by considering such images from different modalities for better explanation and modelling of diseases and cure of diseases. Out of these, T1 and T2 weighted images are the most commonly acquired MR images as they provide the structural information of soft tissue. Thus, it has become a standard to acquire T1 and T2 weighted images in structural sections. However, the sequential acquisition of such images increases the acquisition time which is generally undesirable. In this context, the T2 weighted images require a much longer acquisition time (AQT) as compared to T1 weighted images [31], which makes the acquisition procedure longer and thus prone to patient discomfort and also hampers the speed of MRI availability. The patient discomfort may also result in increased motion artefacts in acquired images, and fewer scans per scanner increases the financial burden. Many acceleration methods for MR acquisition rely on undersampling the k-space, but suffer from a trade-off between acquisition time and quality of MR images. For instance, acquisition of T1 weighted and T2 weighted images takes ∼10 minutes and it may take ∼4-6 minutes for undersampled T2 weighted k-space (1/8 samples) along with T1 weighted, but the quality of such T2 weighted images are too low to be used for diagnostic purposes.

Various alternatives (explored so far) to reduce the acquisition time for such modalities can result in a decrease in spatial resolution and a decreased number of signal averages, leading to poor quality of acquired images. Hardware solutions in this direction address the designing of new pulse sequences that can synthesize different modalities in a single shot of scan. However, installing new scanners with these pulse sequences is often too expensive to be deployed under a typical Indian scenario. These issues demand exploring a new direction of developing post-capture algorithms which can efficiently and reliably map an image from one modality (e.g. T1 weighted) with or without the undersampled version of T2 weighted to the other (T2 weighted), making the need to acquire the T2

weighted image oblivious. Our proposed work takes ∼1 second to reconstruct a complete T2 weighted volume. Thus reducing the acquisition time to increase the patient's comfort and decrease the cost per procedure is the prime focus of this chapter.

3.1.1 Motivation and Problem Statement

Unlike many of the existing methods, the focus of this chapter is to reconstruct T2WI from the given T1WI without any requirement for anatomical prior information of the subject. The hypothesis behind such an intermodality reconstruction task is that the underlying source to be imaged is the same. Its reconstruction involves learning a mapping between two representations of this source (e.g. T1 and T2). Here, the structural properties of source and target modality images are known from the training data used in the algorithm. These properties are assumed to be similar across the subjects in the case of MR images. The motivation behind learning such a mapping is that information in different types of MR images with different contrast from a single subject is highly co-related, as shown in Figure 3.1. In T1WI white matter appears bright but is darker in T2WI. Similarly, grey matter is dark in T1WI and appear bright in T2WI. Apart from such normal contrast changes, several pathologies can appear iso-intense (same as surroundings) in T1 images but will be hypo (dark)/hyper (bright) in the T2 images. If pathology is hypo-intense in T1WI and hyper-intense in T2WI, they contain similar information, i.e. normal contrast changes in T1 and T2 images. An example of such lesion is shown in Figure 3.1 from dataset [29]. The lesion in the blue box appears hypo in T1 and hyper in T2 weighted images, thus conveying similar information. However, the red ellipse lesion is iso-intense in the T1 weighted image but hyper in the T2 weighted images.

Authors	**Work**
Raviteja et al. 2015 [32]	First initiative for unsupervised reconstruction of intermodality MR images. Use cross-modality nearest neighbour search to select one candidate out of multiple target modality candidates with maximum mutual information. Dataset Used: NAMIC. Correlation (T1/T2) =0.839, SNR (T1/T2) =12.78
Cagan Alkan et al. 2017 [33, 34]	Analysis of learning of convolutional network with 9 layers used for reconstruction of MR intermodality images in presence and absence of depth, tissue masks. Dataset Used: HCP dataset [30]. Network loss with tissue mask=7.728 and without tissue mask=9.093.
Yawen Huang et al. 2017 [35]	Learn sparse based dictionaries in latent space, the transformation among coefficients and across image modality matching criterian is proposed. It requires weakly coupled MR images. PSNR is 34.27 on IXI dataset and 30.40 on NAMIC dataset.
Xiang et al. 2018 [3]	First initiative to use undersampled k-space information and learned transformation using DenseNet. Dataset used: MICCAI challenge [29]. PSNR with only T1 is 30.60, with T1 and 1/8 T2 is 36.90
Agisilaos et al. 2018 [36]	Learns to embed all input modalities into a shared modality-invariant latent space. These latent representations are then combined into a single fused representation, transformed into the target output modality with a learnt decoder. Dataset used: ISLES and BRATS.
Sharma et al. 2019 [37]	MM-synthesis is a variant of a generative adversarial network (GAN) that is capable of leveraging redundant information contained within multiple available sequences in order to generate one or more missing sequences for a patient scan. Dataset used: ISLES-2015 and BraTS-2018. PSNR of BraTs LGG is 24.24±2.46 and of BraTs HGG is 24.78±2.89
Dar et al. 2019 [38]	pGAN preserves intermediate-to-high frequency details via an adversarial loss, and it offers enhanced synthesis performance via pixel-wise and perceptual losses for registered multi-contrast images and a cycle-consistency loss for unregistered images.

Table 3.1: Related works in the literature addressing reconstruction of MR intermodality images. LGG and HGG stand for Low/High grade Glioma.

3.1.2 Related Works

Some recent works reported addressing such a task of reconstructing the images describing the source from the different images of the same source. These approaches model the reconstruction problem as learning the transformation between spaces spanned by acquired modality and target modality images. The space as well as transformation is learned in various frameworks based on sparse representation and neural network models [3, 35, 39, 40]. A generalized framework for synthesizing the PD weighted and T2 weighted images has been reported in [35] via learning the transformation along with dictionaries in the sparse representation framework. An unsupervised method to reconstruct the T2 modality images from T1 weighted images is proposed in [32] which works by selecting the best candidate from multiple target modality candidates via maximizing a global mutual information cost function keeping the spatial consistency.

On the other hand, deep learning techniques have become the centre of attraction for developing machine learning solutions in many areas. In MRI, the accelerated reconstruction of images as well as the prediction of contrast for input images has been explored using deep learning techniques [33, 34]. The feasibility of constructing different contrasts from a single MR image modality using convolutional networks is experimentally verified with and without the tissue labels as input features in [33]. Considering the importance of complementary information present in different modalities, few samples of k-space for T2 weighted images, along with the complete T1 weighted images, are, for the first time, motivated to be utilized in the construction of T2 weighted images from given T1 weighted images using DenseNet [3]. A tabulated literature summary addressing reconstruction of MR intermodality has been reported in Table 3.1.

3.2 Proposed Method

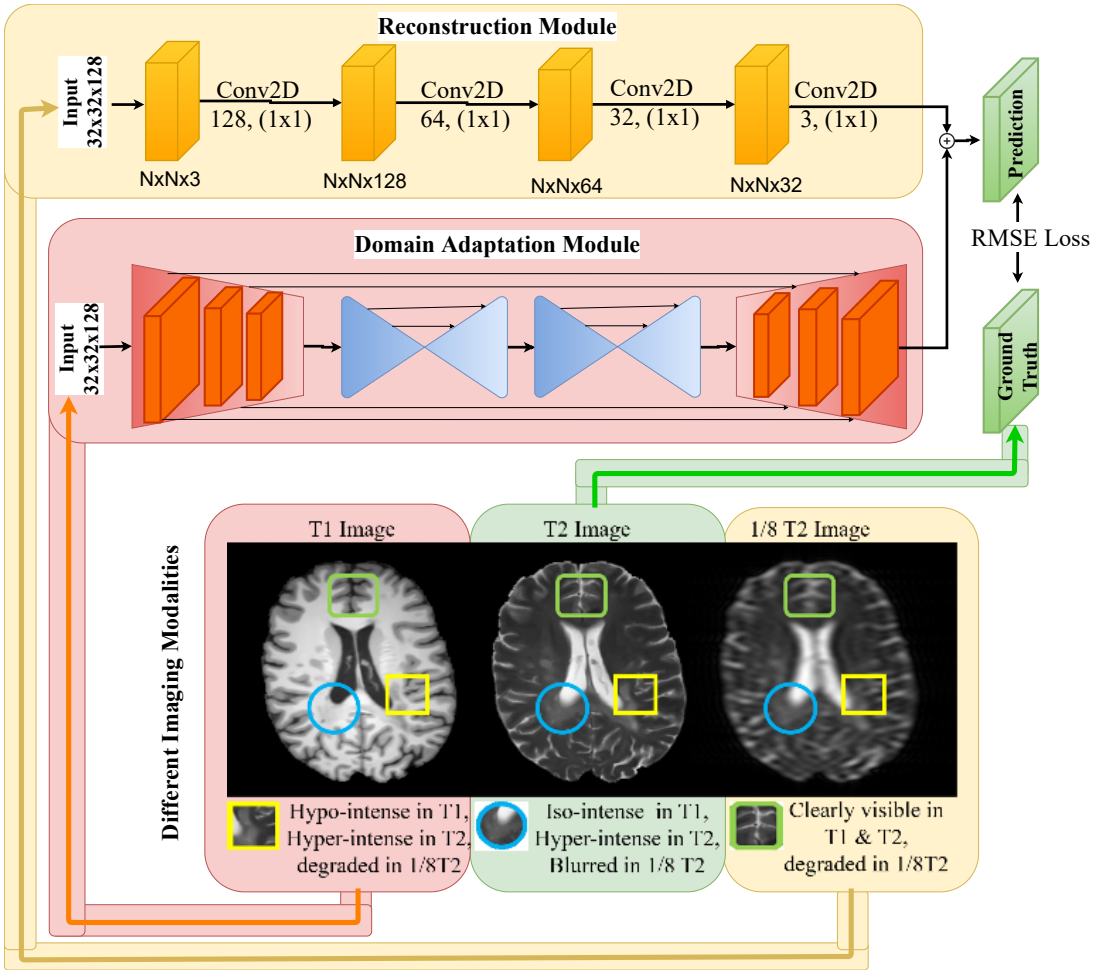


Figure 3.2: Illustrations of T1 weighted image, T2 weighted image and under sampled version of T2 weighted image. The architecture layout of proposed approach is shown in above row.

The proposed network can generate T2WI of good quality. It is adapted for MR images via effective modular training and incorporating gradient information of the image in two orthogonal directions (x and y) along with image intensity values, leading to a three multi-channel loss used for regularized network convergence. The network imitates contraint learning by utilizing undersampled k-space of T2WI as constraint to learn better image details, especially the iso-intense details in T1WI. The T2WI formed by a few k-space

samples are fed to the Reconstruction Module (RM) in parallel to the DAM, and the outputs of both RM and DAM modules are finally added for computation of T2 weighted image at the output of proposed work as shown in Figure 3.2. This ensures that the DAM module learns the residual information accurately. The proposed approach outperforms the existing algorithm [3] in qualitative as well as quantitative measures. The rest of the chapter discusses major components of the proposed network, loss function, architecture details and training procedure, experiments signifying advantages of proposed work, data preprocessing details, analysis of reconstruction qualitatively, and quantitatively followed by a summary concluding the chapter. The major contributions of this chapter are

- Introducing Reconstruction Module (RM) to incorporate the undersampled k-space information of T2 weighted image to aid the DAM network for better learning.
- Image gradients in two orthogonal directions are fused as the input and ground truth for defining multi-channel loss, used for regularized network convergence.
- The vanishing gradient problem is dealt locally in the network using residual networks, and the stacked SBMs help in feature extraction at multiple scales.
- The proposed approach outperforms the existing algorithm [3] in qualitative as well as quantitative measures.

Since the quality of MR images is subjectively defined by clear tissue boundaries evaluated at various scales and a high SNR, it is important to process the brain MR image at multiple scales, i.e. processing of micro and macro image details. Further, to provide a realistic solution for the reconstruction of the T2WI, we focus on a simpler network design in this chapter which imposes a less computational burden and yet can provide better reconstruction quality. The proposed network consists of two major modules: Domain Adaptation Module (DAM) and Reconstruction Module (RM). The DAM reconstructs the

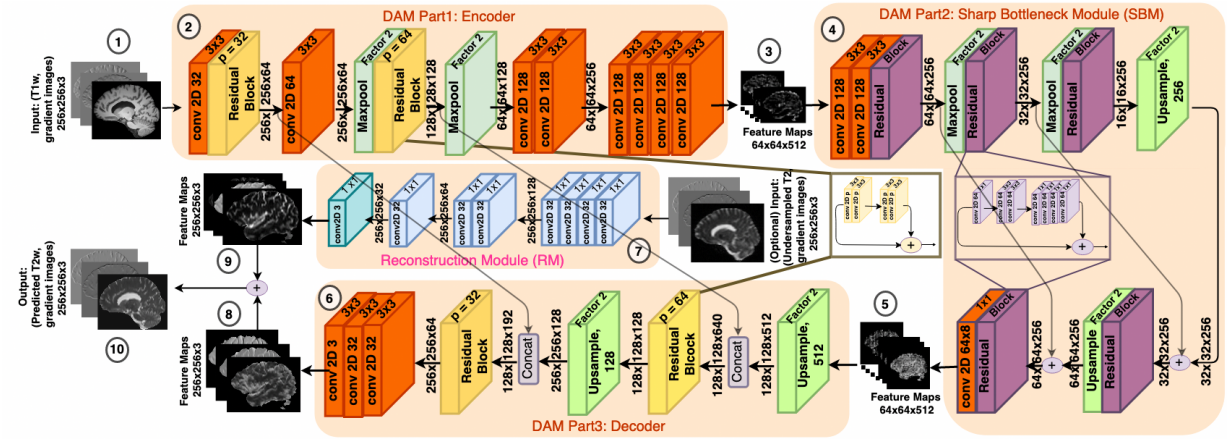


Figure 3.3: Architectural details of proposed network (Zoom for better visualization). DAM Part 1 is called the encoder, and it downsamples the data to a bottleneck. DAM Part 2 is called the SBM, and it captures the features on a global scale. DAM Part 3 is called the decoder, and this reconstructs the required image. RM improves the reconstruction by processing under-sampled T2WI. Order of layers in the network can be followed by the circled integers 1 to 10.

T2 image from the corresponding T1 image using an encoder-decoder model. In order to generalize and enhance the learning ability of the Encoder-Decoder model, the features encoded by the Encoder (Figure 3.3(3)) are further downsampled to a very low dimensional space and then upsampled to learn the transformation at a finer scale as step 5 in Figure 3.3. Such downsampling is done using stacked Sharp Bottleneck Module (SBM) included in the DAM.

Further to avail the incorporation of under-sampled T2 weighted k-space samples in better transformation learning, the T2WI is passed through a few convolutional layers (discussed in detail in Section 1.2.2), named as Reconstruction Module (RM), till the output of DAM module as shown in step 8 of Figure 3.3. Such incorporation of RM helps avoid the vanishing gradients and aids the DAM module for better learning of transformation. It can result in a T2 weighted image, which is essentially more accurate than the T2 image at the input of RM. Finally, the output of both RM and DAM modules is concatenated in the third dimension as shown in 8 and 9 of Figure 3.3 to produce the output in the proposed

network. The predicted T2 weighted (along with two gradient images) and ground truth T2 weighted image (with two gradient images) are compared using root mean square error (RMSE) loss for image similarity and is computed only on the non-zero pixel values to emphasize the error only for brain region and to avoid. This is also because the background pixels, being zero-valued, lead to a huge decrease in loss which can result in vanishing gradients. Loss is backpropagated through the concatenation layer into both the DAM and RM to improve learning. We discuss the salient aspects and justification of each unit of the proposed approach in the following sections.

3.2.1 Domain Adaptation Module (DAM)

In this subsection, an Encoder-Decoder network is embedded with SBM modules and is termed as Domain Adaptation Module (DAM). To learn the transformation at multiple scales, a Stacked Bottleneck Module (SBM) is derived from hourglass network [41]. **Encoder :** The encoder network Part 1 of DAM in Figure 3.3 takes the input T1 image of size $256 \times 256 \times 3$ and uses 2-D convolutional layers for extracting a $64 \times 64 \times 512$ feature maps as shown in step 3 of Figure 3.3. The encoder consists of blocks of 2D convolutional layers with filters of size 3×3 followed by a residual block and Max-Pool layers. After the residual block, another layer of filter size 3×3 has been applied, and then maxpooling of 2×2 is performed. The shrunk feature map is passed through another residual block followed by maxpooling. Finally, two convolution layers having 3×3 filters are applied to get a feature map of shape $64 \times 64 \times 512$.

SBM : In SBM module Part 2 of DAM in Figure 3.3, the encoder part takes the feature map of size $64 \times 64 \times 512$ and shrinks it down to a small feature representation of $16 \times 16 \times 256$ using residual blocks and maxpooling layers. The decoder takes the $16 \times 16 \times 256$ feature maps and reconstructs the feature map of size $64 \times 64 \times 512$ using residual blocks and

upsampling as shown in step 5 of Figure 3.3.

Decoder : Similar to the encoder, the Decoder Part 3 of DAM in Figure 3.3 is also a 2-D Convolutional network that reconstructs the T2 image from the transformed feature map obtained from SBM. It consists of residual blocks having a Transpose Convolutional layer of filter size 3×3 to increase the feature map's size. The network architecture of the decoder is shown in Figure 3.3. The decoder takes the $16 \times 16 \times 512$ feature maps and constructs the final T2 output of size $256 \times 256 \times 3$ as shown in step 8 of Figure 3.3.

3.2.2 Sharp Bottle Neck Module (SBM)

The ability of a network to learn a better transformation between input and output image depends upon the learning of the features at a variety of scales. The features learned for image reconstruction problems should be able to represent the images at different scales, so the mapping is learnt between all scales of features. The global features are learnt at a macro (coarse) and local features at a micro (fine) scale. Global features of images can be maximum image size, and finer-scale features can be as small as pixel size. However, there is a trade-off between the fine-scale limit and image details loss. For example, in MR images, many major and minor image details exist that play a vital role in clinical diagnosis.

Max-pool functions are used in the encoder-decoder network, which enables the down-sampling of images to provide features at various scales while preserving the semantic information and removing redundant values. SBM, which downsamples and upsamples the features, is arranged in a cascaded fashion one after the other to re-evaluate the significance of the obtained global features after each SBM. Architecture can be seen more elaborately in Figure 3.4 than in Figure 3.3. The proposed stacked SBM structure is inspired by the

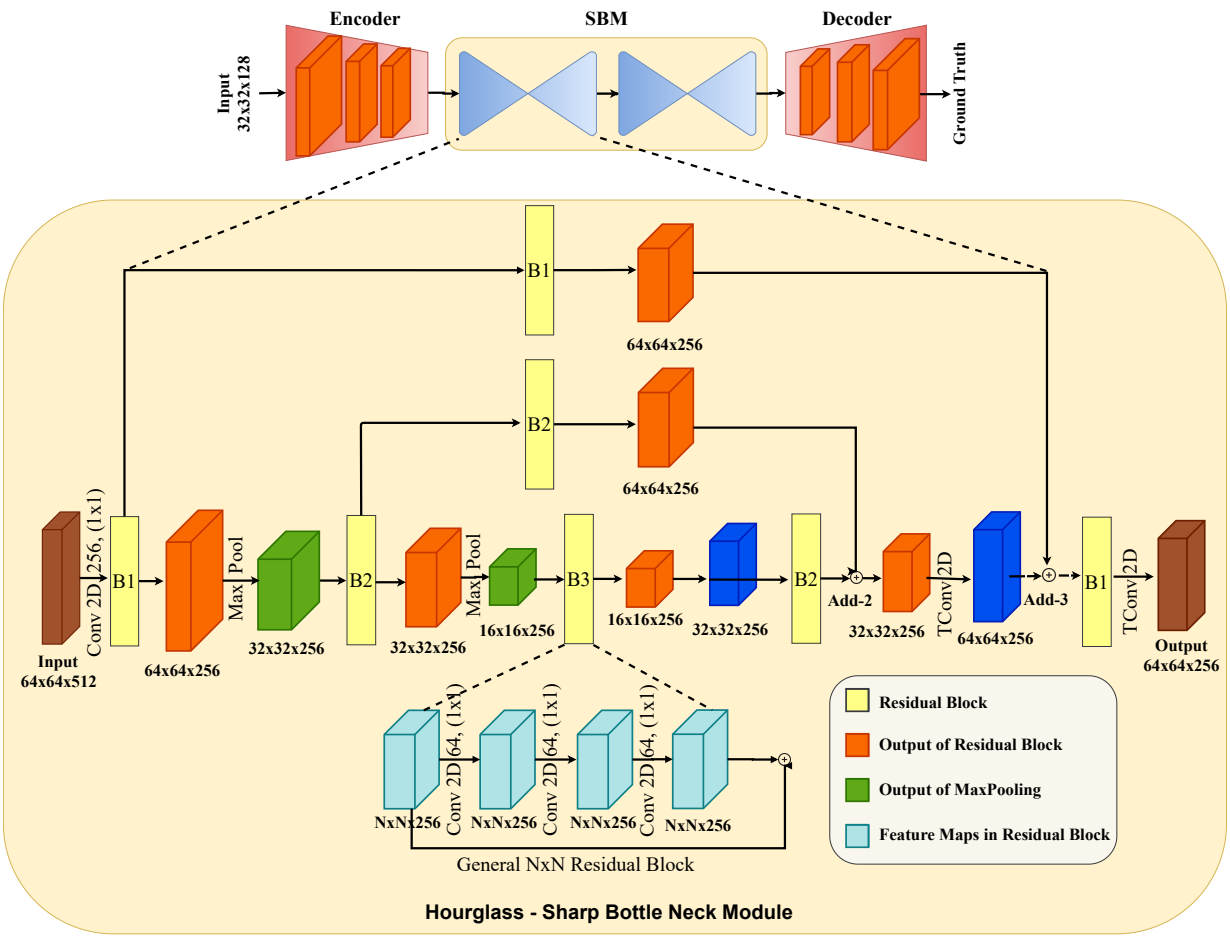


Figure 3.4: Network architecture of Sharp Bottleneck Module (SBM) which is shown as a part of Domain Adaptation Module (DAM).

idea of repeated bottom-upsampling the images to obtain better features [41]. It has been observed to introduce more non-linearity by churning out better multi-resolution features. However, the proposed work differs from [41] as the number of bottleneck modules is restricted to 2 instead of multiple as in [41] because MR images possess a largely similar structure. Thus superior features can be obtained using two bottleneck modules themselves. Also, it has been seen in experiments, SBM tends to learn the finer-scale details than just the encoder, as shown by activation maps in step 3 and step 5 of Figure. 3.3.

Further, the second SBM helps in smooth convergence, and thus more SBMs as in [41] is redundant for this application. Further, to estimate the T2W without any loss of

image detail information in SBM (due to downsampling), skip connections are used to connect the encoding and decoding part of SBM. Residual connections are utilized in the encoding and decoding part of the DAM. They help retain information at the global and local features, respectively, thus allowing easy flow of gradients.

3.2.3 Under sampled k-space

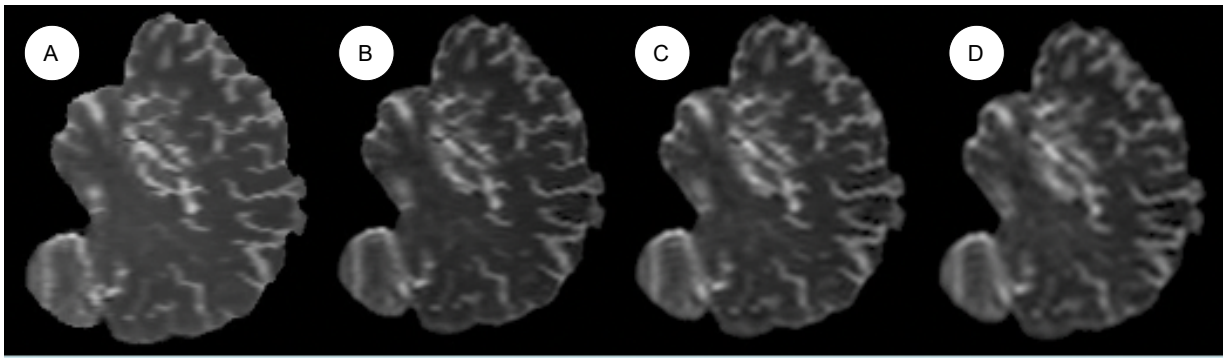


Figure 3.5: A: T2WI with fully sampled k-space, B: Under Sampled T2WI with 1/4th sampled k-space, C: Under Sampled T2WI with 1/8th sampled k-space and D: Under Sampled T2WI with 1/16th sampled k-space

Apart from the reconstruction of T2WI from T1WI, experiments are also carried out to reconstruct the T2 weighted images using T1 and the T2 weighted priors (used in the RM) generated from a few k-space samples of T2 weighted images. The undersampling procedure in this work is adopted from existing works [3] for a fair comparison. The k-space samples are first undersampled retrospectively, choosing a fraction of samples from central low-frequency components. For example, 1/8 T2W means the T2WI is generated from k-space only with $\frac{1}{8}^{th}$ low frequencies sampled from the central part of k-space. The 1/8 T2W used in the proposed work has **31.38dB** PSNR, which is lower than 1/8 T2W used in [3] with **32.4dB** and thus can be used for a fair comparison. Detailed comparison Undersample T2 generated with various fractions of k-space is shown in Figure. 3.5.

3.2.4 Reconstruction Module (RM)

Although using only the DAM yields a good quality reconstruction of T2WI, it does not use any prior information about the appearance of T2WI. To use such prior information (optionally, if undersampled T2W information is available), DAM is augmented with an RM as shown in Figure 3.2. Thus, RM can play a role in regularizing the optimization of DAM to learn a better transformation. As a prior, T2W image formed by the undersampled k-space samples is provided at the input of RM and its output is connected with the output of DAM module. The idea behind using the undersampled T2WI as a prior via RM instead of direct connection is that the convolutional layers of RM tend to learn the better reconstruction of T2WI at its end, as be seen in the activation maps of Figure 3.3. As the output of RM and DAM modules are connected, the RM part reinforces the DAM to learn the transformation better, as the final estimate of the T2WI is now not just via the DAM but is also constrained via the RM.

3.2.5 Multi Channel Input

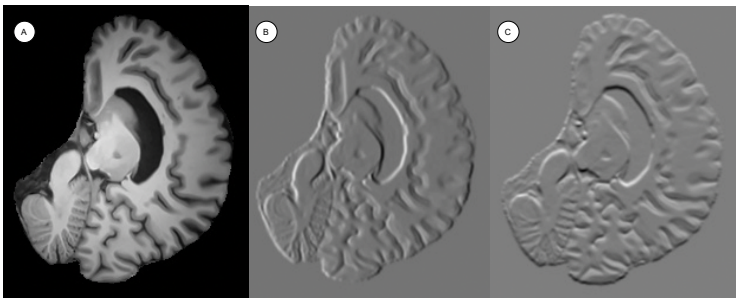


Figure 3.6: A: Image, B: Gradient of A in 'x' direction, C: Gradient of B in 'y' direction.

Edges represent important details in images and its benefits to explicitly consider them while addressing the reconstruction problem. Indeed, both T1 and T2 weighted images are differentiated by the contrast of different tissues, so the sharpness at the boundary of tissues is important for good localization of the structural details and therefore must be preserved.

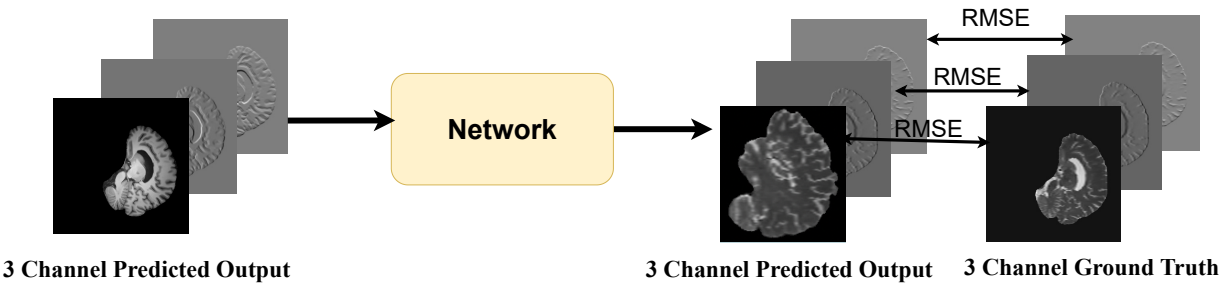


Figure 3.7: Loss is computed across all channels.

This is addressed using a three-channel input, i.e. one channel incorporates intensity T1 weighted image, and other two channels incorporate the gradients of the first channel in two orthogonal directions (x and y) as shown in Figure 3.6. The loss is computed over all three channels, and thus the network estimates the intensity T2WI, which has similar gradient profiles as of original T2WI (provided while training) as shown in Figure 3.7. In this way, the network is forced to implicitly learn the transformation, which satisfies such mapping between the gradient images.

3.2.6 Residual Blocks

There are multiple convolutional layers in the entire network, including Encoder, SBM and Decoder. An increase in such layers helps to learn the transformation efficiently by learning high-level features. For example, vanishing gradient problem is avoided, and better learning is enabled in such a network by using a standard technique, i.e., skip connections in SBM and local skip connections between convolutional layers in the three major parts of DAM, i.e. Encoder, SBM and Decoder. This also helps such deep networks to avoid over-fitting. Figure 3.4 shows the architecture of Residual Blocks in detail.

3.2.7 Loss Function

For any T1 image, the input to the network comprises of the 2D T1 image I^{t1}, its normalized 2D X-gradient X^{t1} and normalized 3D Y-gradient Y^{t1}. The network predicts the T2 image, as well as the X-gradient and Y-gradients, represented as $\hat{I}^{t2}$, $\hat{X}^{t2}$ and $\hat{Y}^{t2}$ respectively. For training the network, loss is computed between each pixel of the predicted output and the corresponding ground truth 2D T2 image I^{t2} and its X-gradients X^{t2} and Y-gradients Y^{t2} as shown in Figure 3.7. The loss function used is

$$L = \sqrt{\frac{1}{N}\sum_{j=1}^{N}[(I_j^{t2} - \hat{I}_j^{t2})^2 + (X_j^{t2} - \hat{X}_j^{t2})^2 + (Y_j^{t2} - \hat{Y}_j^{t2})^2]} \tag{3.2.1}$$

where N is the number of images in the given batch so $\frac{1}{N}$ averages over all the N images in the batch. This loss is computed over only non-zero pixels to avoid the interference of zero background pixels. Since most of the boundary pixels are black, leading to lesser overall loss, the network could be misdirected to learn the unnecessary information. RMSE measures how far the predicted pixel values are from the ground truth pixels. It can be seen as Euclidean distance scaled by a factor of N. It helps the model estimate how far off from the ground truth it should predict the next time.

3.2.8 Training Procedure

The network is adapted for MRI via effective modular training. For efficient training, a sequence of steps as shown in Step A, B and C as in Figure 3.8 is described below in detail:

- A: First, only the Encoder and Decoder models of DAM are trained for reconstructing T2WI from only T1WI. For the case where undersampled T2WI is put through RM, Encoder, Decoder, and RM are trained in the first step.

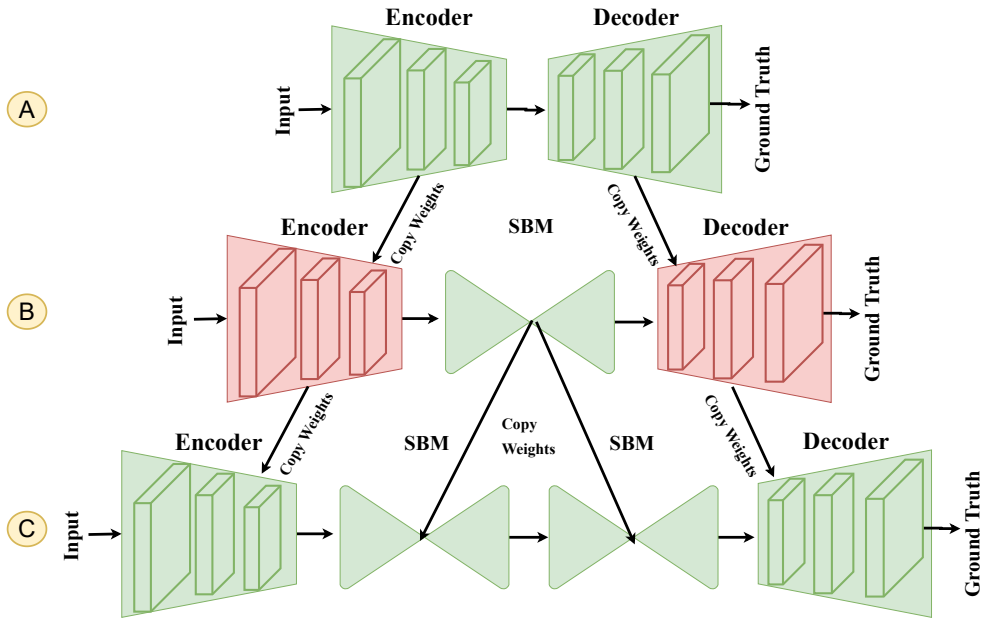


Figure 3.8: A: Train only Encoder and Decoder, B: Add an SBM, copy and freeze the weights of Encoder and Decoder, train only the SBM, C: Copy the weights as-is from step B and make an another copy of SBM and train end to end.
Green region indicates weights which undergo training and Red region indicates weights that are frozen. In case where undersampled, RM is augmented in Step A to the network.

- B: One SBM has been added between the trained Encoder and Decoder. The weights of trained Encoder and Decoder (RM also in alternative case) are loaded and frozen. Then, only this SBM is trained. However, the loss back propagates through all the three parts (Encoder-SBM-Decoder). Only the weights of the SBM will be updated according to the gradients wrt the loss in this step.

- C: Finally, a copy of the first SBM is made, and two of them are stacked between Encoder and Decoder to form the full DAM. Now, the weights of Encoder-Decoder from Step 1 are loaded, and weights of SBM from step 2 are loaded on itself and its copy. Loss is back-propagated, and weights are updated accordingly in all four pieces (Encoder-SBM-SBM-Decoder) and all the five pieces in case of DAM with RM. Training is done in an end to end fashion in this step.

Benefits of modular training are: High dimensional function being learned rely on the

number of learnable parameters in the neural network. A large number of parameters makes the learning procedure difficult to reach the minima. However, in our case, learning a complex transformation involves many parameters (∼6.9M) and is observed to be complex to optimize. So the parameters are learned in parts at first to achieve reasonably good quality piecewise solution and then fused to improve the solution further.

3.3 Metrics Used

To understand how good the performance of the system is, we should be able to measure it. An objective score between the ground truth and predicted value will indicate the performance. The score can indicate both the quantity and perceptive quality of the performance. There are also a variety of measures to understand different aspects of the system.

3.3.1 Quantitative Performance Metrics

In the following equations, I_j means the Ground truth image and $\hat{I}_j$ means the Predicted image.

- **Mean Absolute Error (MAE):**

$$Loss = \frac{1}{N}\sum_{j=1}^{N}[(I_j - \hat{I}_j)] \tag{3.3.2}$$

  MAE measures the average magnitude of error of the test sample. Absolute value makes sure that the direction is not taken into consideration.

- **Mean Square Error (MSE):**

$$Loss = \frac{1}{N}\sum_{j=1}^{N}[(I_j - \hat{I}_j)^2] \tag{3.3.3}$$

MSE is always non-negative, and values closer to 0 indicate better performance. It is averaged over the entire test sample. Small differences are exploded due to the squaring, so error tolerance is less compared to MAE.

- **Peak Signal to Noise Ratio (PSNR):**

  It is the ratio between the maximum possible value and the corrupting noise that effect. Higher PSNR usually indicates that the reconstruction is of higher quality.

$$PSNR = 10 * \log_{10} \frac{MAX_I^2}{MSE_{I,K}} \tag{3.3.4}$$

where I is the noise free image, K is the predicted image, MAX_I^2 represents the maximum value the image I can take and $MSE_{I,K}$ is the MSE between I and K. In the absence of noise leading to zero MSE, PSNR tends to become infinite.

3.3.2 Similarity Measure

Quantitative measures like the ones above do not always tell enough about the reconstruction. Scoring the perceptual quality of the image as visible to the human eye needs a different kind of metric altogether. MSE and PSNR do not consider the dependency between the pixels located spatially close to one another. This does not allow them to measure the perceptual quality.

Structural Similarity Index Measure (SSIM):

SSIM is a perception based model [42]. Structural information is based on the fact that spatially close pixels have a huge interdependence. This dependence tells a lot about the scene and texture of the image. Thus, SSIM is based on luminance, contrast and structure.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{3.3.5}$$

- x and y are the two windows of same size being compared
- μ_x and μ_y are the mean of x and y
- σ_x, σ_x are the variance of x and y and σ_{xy} is the co-variance of x and y.
- $c_1 = (k_1, L)^2$ and $c_2 = (k_2, L)^2$ are two variables to stabilize the division where L is the dynamic range of pixel values. It is typically $2^{(\#\text{ bits per pixel})} - 1$
- $k_1 = 0.01$ and $k_2 = 0.03$ by default.

3.4 Datasets Used

To demonstrate the performance of the proposed work, experiments are performed on two different publicly available datasets with real MR images-

- Dataset1 : T1-T2 weighted paired images for 5 subjects [29] with spatial resolution $336 \times 336 \times 261$.
- Dataset2 : We randomly chose 45 subjects for training, 6 subjects for validation and 6 other subjects for testing from the data in [30]. The spatial resolution of T1-T2 weighted paired images in this case is $260 \times 260 \times 311$.

For Dataset-I, the leave-one-out cross-validation technique is used to evaluate the generality of the learned transformation due to the lack of enough data for testing. For Dataset-II, as the number of subjects is enough to incorporate the variability - results were produced on the test subjects. Since the state of the art [3] demonstrates the results only for Dataset-I,

so the comparison of the proposed work with the existing method is made only for this dataset. Moreover, networks with only DAM, and when using DAM and RM modules, are compared to gauge each module's significance. The 2D axial slices of MR image volumes are used for training. The initial and last 30 axial slices are discarded due to insignificant brain tissue available in scan volume. Most parts of the corner 2D slices are black with very few brain region pixels in the centre. As mentioned in the dataset [29] as a part of preprocessing the data, the brain region of each image is extracted using a brain extraction tool (BET) in FSL software [43]. Further, to have the pixel to pixel correspondence, the T1 weighted images are registered with the corresponding T2 weighted images using the FLIRT algorithm in FSL [44].

3.5 Experimental Results

The reconstruction quality of MR images is evaluated quantitatively by three evaluation metrics: (i) peak signal to noise ratio (PSNR), (ii) mean absolute error (MAE), and (iii) structural similarity index (SSIM). As only MAE and PSNR are used in existing work [3], to quantitatively compare the proposed approach with work in [3] these two metrics are used. However, all three metrics are used for the quantitative evaluation of the proposed work for Dataset-II. It has to be noted that the proposed work is compared with only [3] since only this approach has used the partial k-space samples for the reconstruction of T2 weighted images. As pix2pix [45] is a popular image-image translation method, comparisons between the proposed approach and pix2pix are provided for the "with only T1WI" as input. Existing literature has used Dataset-I for comparative analysis with existing work. Following the existing approach [3], the T2 images are reconstructed in the proposed work using two approaches shown in the following subsections as Experiment 1 and 2.

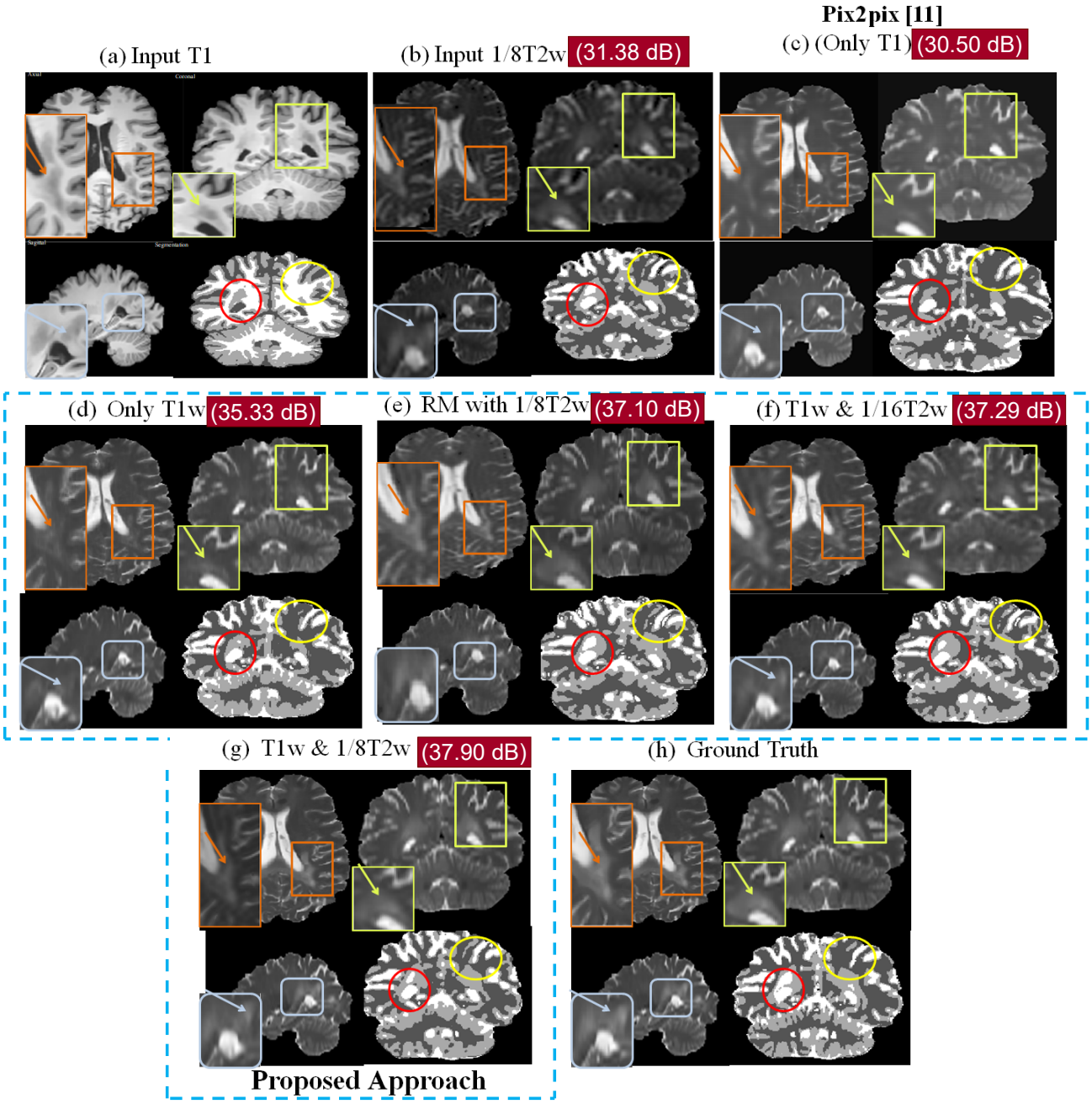


Figure 3.9: Blue dotted line encompasses the results from the proposed network. Each block shows views of the output image in (clockwise) Axial plane, Coronal plane, Segmentation map of Coronal plane, Sagittal plane. (a) Input T1WI, (b) Input 1/8th T2WI whose PSNR with T2WI is 31.38 dB. The process of estimating this image is given in Section 1.2.3. (c) Predicted T2WI ($\hat{T}_2$) with only T1WI using Pix2Pix network, (d) $\hat{T}_2$ with only T1WI using only the DAM part of the proposed network, (e) $\hat{T}_2$ with only $\frac{1}{8}$th T2WI using only the RM part of the proposed network, (f) $\hat{T}_2$ with T1WI and $\frac{1}{16}$th T2WI using the proposed network, (g) $\hat{T}_2$ with T1WI and $\frac{1}{8}$th T2WI using the proposed network, (h) Ground Truth T2WI.

Metric	Method	Reconstructed T2 with only T1	Reconstructed T2 with 1/8 T2 and T1	Reconstructed T2 with 1/16 T2 and T1
PSNR	DenseNet [3]	30.60	36.90	34.3
	Proposed	**34.07**	**37.30**	**36.50**
MAE	DenseNet [3]	33×10^{-3}	14×10^{-3}	19×10^{-3}
	Proposed	$\mathbf{5.01\times10^{-3}}$	$\mathbf{3.46\times10^{-4}}$	$\mathbf{3.87\times10^{-4}}$
Data-II	PSNR	32.13	33.08	32.96
	MAE	7.81×10^{-3}	6.86×10^{-3}	7.13×10^{-3}

Table 3.2: Quantitative comparison with existing work [3].

3.5.1 Experiment 1: Reconstruction of T2WI from only T1WI

Reconstruction using only T1WI. It involves input with only T1WI, and the network used to reconstruct is DAM only.

Quantitative comparison with existing work : Comparative analysis is performed by evaluating PSNR and MAE with [3] for five subjects reported in Dataset-I. It can be observed from Table 3.2 that in the reconstruction of T2WI using only T1WI, our proposal has higher PSNR than the reconstruction performed in [3]. The mean value of the evaluation metrics across volumes from Dataset-II is mentioned in stand-alone Row 2 of Table 3.2 below the results of Dataset-I because previous work [3] has shown results only for Dataset-I. We can note that the MAE and SSIM values are close to their minimum (maximum) limits and thus indicate an accurate reconstruction of T2 images. Comparisons between our approach and pix2pix [45] are provided for "only T1WI" as input because there was no possibility to provide an undersampled image as guidance. The best PSNR obtained using ***pix2pix [45] is 30.1dB*** which is lesser than PSNR of the proposed work which is ***34.07dB***.

Qualitative analysis: The qualitative analysis of reconstructed images shall be done in two ways: check the reconstruction quality of image details (i) present in T1 weighted image and (ii) not present in T1 weighted image, but are present in original T2 weighted

image. The reconstruction results of the proposed work are compared with the style transfer algorithm for images called pix2pix [45] and with [3]. The proposed work addresses 2D image processing (in axial plane) inherently assumes independence among slices which is not true for neighbouring slices. However, here the reconstruction in all three planes is shown for a randomly selected image in Figure 3.9 to evaluate the reconstruction quality better visually. In addition to the quality comparison in better estimation of image details and reduced artefacts while reconstruction, we also focus on the utility of reconstructed images for post-processing applications. Thus, segmentation maps for cerebral spinal fluid (CSF), white matter (WM) and grey matter (GM) are shown using three distinct grey levels in the bottom right of each block of the Figure 3.9. The image details with an abnormality are shown in the zoomed window. PSNR values of each of the slices are also mentioned above each block. It can be observed that the image reconstructed by pix2pix [45], which uses UNet architecture in the Generator model, does not reconstruct the abnormal image detail well (in all three planes). Nevertheless, the proposed approach with T1WI input can reconstruct the details better in the axial plane. Also, pix2pix cannot provide accurate segmentation labels and tends to produce artefacts by removing the grey segmentation label (can also be seen in red and yellow circles) compared to the proposed method.

To demonstrate the feature learning ability by the proposed network on other datasets, 3T MRI images from HCP (45/6/6 subjects for training/validation/test) has been used (Dataset-II [30]). The mean values for PSNR, MAE and SSIM are listed in Table 3.2. Furthermore, the MR images are reconstructed for the HCP dataset for generalizability. It has been observed that the proposed approach can reconstruct the MR images with similar quality as in the MICCAI challenge dataset.

3.5.2 Experiment 2: Reconstruction of T2WI using T1WI and information of undersampled k-space of T2WI

Reconstruction using T1 weighted image and T2 weighted image generated from undersampled k-space has been done in this experiment. In order to check the impact of sampling rate, two different undersampling cases are considered, i.e. with rates 1/8 and 1/16. The network used is DAM with input T1WI and RM with undersampled T2WI input to reconstruct the corresponding T2WI. The reconstruction of crucial image details which are not present in T1 but are present in T2 weighted images is enforced by utilizing the undersampled k-space of T2 images. Utilization of information of undersampled k-space samples constraints the network to learn reliable transformation such that it leads to a more accurate reconstruction of T2 images.

Quantitative Analysis : The utilization of the prior (partial T2 weighted image) via the RM module helps to improve the reconstruction of the T2 weighted image and thus provide a boost in the performance wrt PSNR and MAE. Since only [3] has attempted to use the partial information of T2 k-space and is proved to perform better than U-Net, the proposed method's results are compared only with [3]. The quantitative comparison is tabulated in Table 3.2. It has to be noted that the dataset and pre-processing steps are kept minimal as well as same in proposed work to that in [3] because their code is not made public. As mentioned earlier, even the 1/8 T2W used in the proposed network has lesser PSNR as compared to 1/8 T2W in [3]. The proposed work outputs T2WI, which has higher PSNR as compared to [3], signifying better transformation learning by the proposed network. There is a significant performance improvement as shown in the metric values even with only 1/16 undersampled T2 weighted image in the case of Dataet-II [30]. This signifies the efficacy of the proposed algorithm over different algorithms and the importance of the RM module irrespective of different datasets.

Qualitative Analysis : The reconstructed T2 images using T1 and information of undersampled k-space for Dataset-I have been shown in Figure 3.9. It can be observed that the details that are not prominent in the reconstruction using only T1 images have been reconstructed well and are prominent when partial information of the T2 weighted image is used. The perceptual quality of reconstructing these details using only T1 weighted images is not as good as in reconstruction with under sampled k-space samples. Image details comprise complementary information from other modality, and thus RM can better reconstruct such image details as shown in (f) and (g) parts of Figure 3.9. Also, the improvement in reconstruction quality of T2W image, as more information of the undersampled T2W image is incorporated. The zoomed detail in the axial slice shows crucial information which is difficult to perceive in both input T1WI and corresponding reconstructed image as in Figure 3.9(d). However, it is reconstructed well when the degraded version of T2WI is given in addition through RM as in part (g) of Figure 3.9.

3.5.3 Ablation Study

Training Module.	**T1W input**	**T1W+1/8T2W input**
DAM	32.08	33.10
DAM+1HG	31.87	33.13
DAM+2HG	32.16	33.08

Table 3.3: Quantitative Analysis (Ablations). PSNR obtained by DAM wihtout the SBMs is less than DAM with SBMs in the case when input is only T1WI.

The emphasis of each component of the proposed work, i.e., encoder-decoder network, SBM, RM, is evaluated by reconstructing images from respective modules. The obtained PSNR values for the reconstruction using different components are tabulated in Table 3.3. It can be seen that the PSNR obtained by DAM network without SBMs is less than DAM network with SBMs, indicating effective learning by SBMs. Further, it has been observed in experiments that the SBMs helps in smoother convergence, which is otherwise shaky,

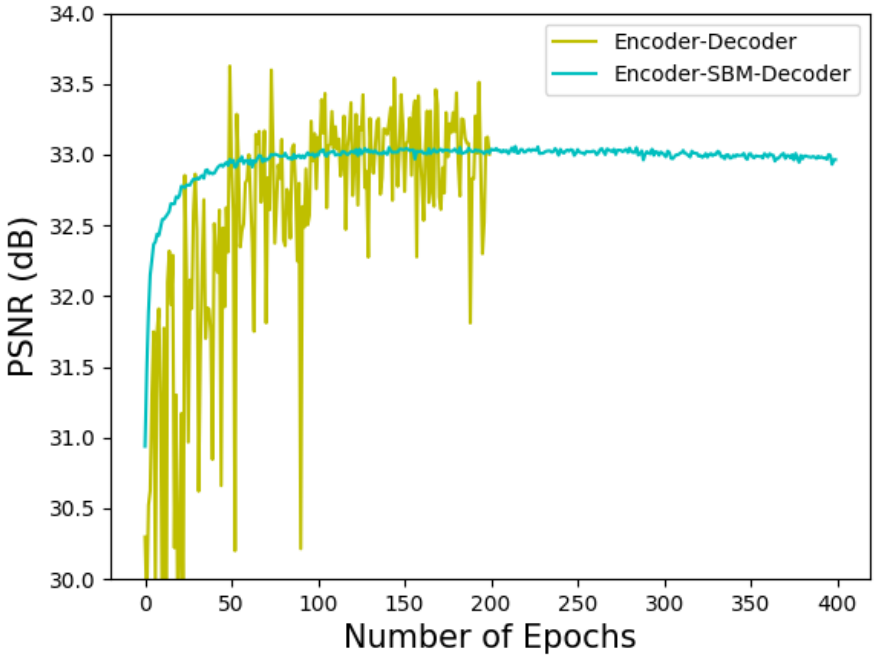


Figure 3.10: Demonstration of significance of SBM module in regularized convergence as well as learning. Fluctuations in training has reduced in Encoder-Decoder with SBM when compared to Encoder-Decoder without SBM.

while learning with only encoder-decoder and RM. To signify the importance of the SBM module and multi-scale processing, the network is trained using only an encoder-decoder network and an encoder-decoder network using the SBM module. The training loss for both the cases has been plotted in Fig. 3.10. It can be observed that the training loss with encoder-decoder network has larger fluctuations as compared to others.

3.5.3.1 Slice wise Inconsistency

In order to check for the inconsistencies across the slices while reconstructing 2D slices in 3D volume data, a pixel in an axial slice is randomly chosen and compared with its variation in the sagittal plane of the original T2WI as shown in Fig. 3.11. It can be seen that the proposed approach can follow smooth changes such as from 100 to 150 slices. However,

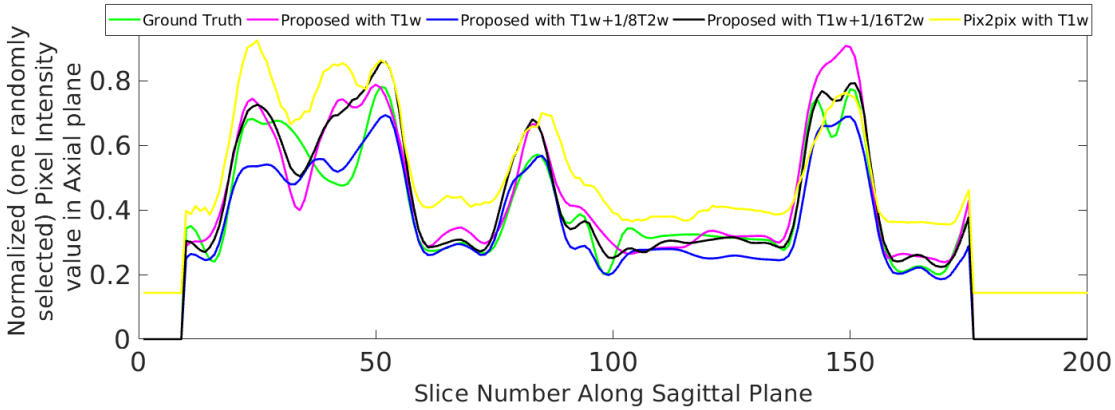


Figure 3.11: Slice level inconsistency while 2D reconstruction

it changes abruptly from slices 0 to 50. Thus, the pix2pix approach provides relatively non-smooth variation overall and does not capture the intrinsic complex structure such as the peak of 150^{th} slice.

3.5.3.2 Segmentation Maps

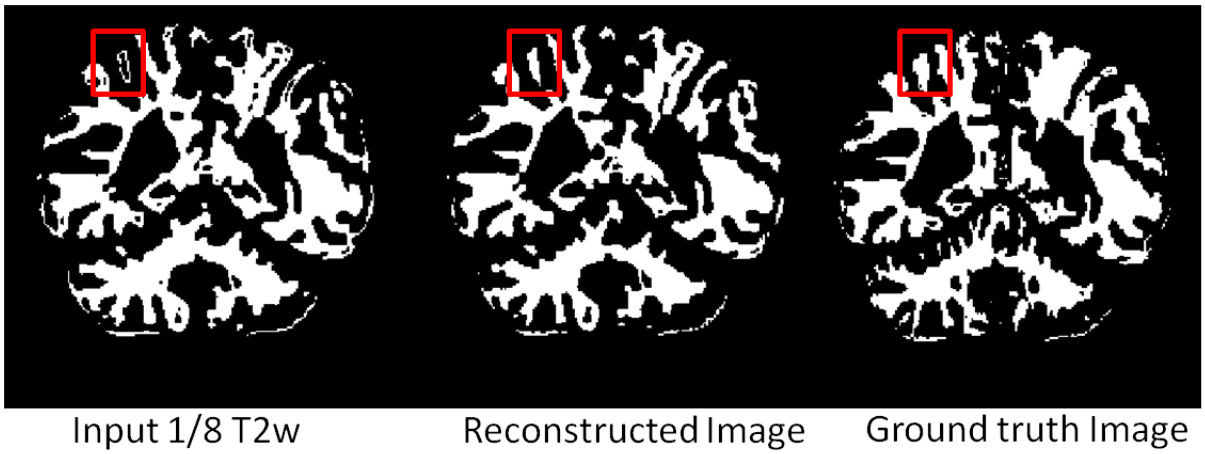


Figure 3.12: Segmentation of white matter as seen in the red highlighted rectangle is improved in reconstructed image as compared to 1/8 T2W image which indicates that the network is able to reconstruct accurately.

White matter is segmented from the images reconstructed by the proposed approach to show the advantages of the proposed reconstruction. Segmentation of [3] is not shown as no

result file is available for [3]. The improved segmentation in one randomly selected reconstructed image using the proposed approach can be seen in the red highlighted rectangle of Figure 3.12, which is otherwise missing in the 1/8T2W image. Such accurate reconstruction of image details leads to better segmentation and aids in better pathological diagnosis.

3.5.4 Run-Time Analysis

All experiments in the proposed work are carried out on a system with Nvidia 1080 TiGPU Xeon e5 GeForce processor with 32GB RAM. The proposed network takes approximately 42 and 46 seconds to reconstruct a 3D T2 volume of a single subject without using any T2 information and with T2 under-sampled k-space, respectively.

3.6 Conclusion

In a structural MR session, it is common practice to acquire T1 and T2 weighted MR images. Out of these, T2 weighted images take a long time. Thus, this chapter discusses a neural network that can reconstruct T2W images from only T1W images and under-sampled T2W images (only if available) by learning the transformation among two spaces. Depending upon the availability, the method also can utilize T2WI constructed using a fraction of k-space samples at the input to facilitate better reconstruction. This network stems from Encoder-Decoder architecture, stacks SBMs between the Encoder and Decoder for scale variability in feature extraction and guided convergence. RM is connected in parallel, which leads to a enhanced reconstruction. While the method yields good quality reconstruction even when using only T1 images as input, it is experimentally verified that the proposed network better utilizes the T2 information from under-sampled k-space

and performs far better than only T1 based reconstruction. The proposed approach is validated for healthy subjects from two different datasets, emphasizing the generalizability of the proposed work. The reconstruction quality of MR images is compared with a recently reported method and is observed to outperform the existing work significantly. The proposed network can also be utilized for FLAIR or any similar reconstruction.

Chapter 4

Medical Image Inference

In the previous chapter, we targeted the benefit of image synthesis in case of MR imaging. In this chapter, we will see the benefits of image inference task in X-rays. Inferring in medical images is referred to as producing preliminary diagnosis from the data. It can be very useful to a radiologist in screening more patients in a given time. More specifically, we consider, constructing report for a given X-ray image.

4.1 Automated Xray Report Generation

Understanding radiology images such as X-Rays is essential for diagnosis and treatment of many diseases. Given the amount of skill required for accurately reading such images [46], it is challenging for less-experienced radiologists to write medical reports. Hence in healthcare, writing medical reports from X-Ray images becomes a bottleneck for clinical patient care. Compared to other image captioning tasks where coherence is the key criterion, medical image captioning requires high accuracy in detecting anomalies and extracting information along with coherence. That is, the report must convey medical facts accurately and unambiguously. X-ray report constitutes of two parts - Findings and Inference. In

addition to the report, each X-ray has tags which are the critical components of the report. In this chapter a deep neural network is proposed to achieve this task. Given a set of Chest X-Ray images of the patient, the proposed network first predicts the medical tags and then generates a readable radiology report. For generating the report and tags, the proposed network learns to extract salient features of the image from a deep CNN and generates tag embeddings for each patient's X-Ray images.

The proposed network uses transformers for learning self and cross attention. Image and Tag features are encoded with self-attention for a finer representation. It uses both the above features for the cross attention along with the input sequence to generate the report's Findings. Then, cross attention is applied between the generated Findings and the input sequence to generate the report's Impressions. Publicly available dataset has been used to evaluate the proposed network. The performance indicates that it can generate a readable radiology report, with a relatively higher BLEU score over SOTA [47]. The code and trained models are available at `https://github.com/s3pi/Xray-to-Report`.

4.1.1 Problem Statement

To aid the radiologists, many researchers are investigating the generation of automatic reports from X-Ray images [47, 48] by formulating the problem as image captioning [49]. Although Xray report generation task looks similar to a generic image captioning task, there are fundamental differences and challenges to report generation. The Xray images contain complex spatial information and the abnormalities present in it are difficult to find requiring subject matter expertise. Beyond everything, reports need to be medically meaningful and clinically accurate. Hence the focus must be on generating clinically accurate reports with reasonably good readability in this work. Figure 4.1 shows one example of the medical report and tags present in the IU datset [50] with the generated report and

Figure 4.1: Shows the actual medical report with MTI tags corresponding to an X-Ray image with the report and tags generated from the proposed network. MTI tags are automatically generated. They are the critical components of the report which capture the essence of the diagnosis.

tags from our proposed system. Every aspect of the proposed methodology is designed to tackle the challenges present in automatic report generation.

The IU X-ray dataset [50] is used to perform our experiment. Each report in the dataset corresponds to a patient and there are a variable number (N) of X-Ray images of each patient. In the rest of the chapter, Pid_{img} refers to a set of (N) X-Ray images corresponding to a single patient id. Automatically generated tags from the report represent most of the critical components of the report. Findings and Impressions together constitute a report. Tags are identified for each patient, and its embeddings are used in the report generation along with image features. The two parts (Findings and Impressions) of the report are generated sequentially.

4.1.2 Related Works

An automatic understanding of Radiology images, especially X-ray images, is a well-studied problem. To facilitate that, Wang [51] proposed a large scale dataset for detection and

localization of thoracic diseases from X-ray images. Yao [52] and Rajpurkar [53] proposed using deep learning-based algorithms for efficient detection of various diseases from chest X-ray images. Later works extended the problem by attributing 'texts' like tags and templates to the x-ray images. Kisilev [54] build a pipeline to predict the attributes of medical images. Shin [55] adopts a CNN-RNN based framework to predict tags (e.g., locations, severities) of chest x-ray images. Zang [56] aimed at generating semi-structured pathology reports, whose contents are restricted to few predefined topics.

However, the first work that successfully generated an automatic medical report from X-ray images was proposed by Jing [47]. They proposed to use a hierarchical LSTM based recurrent model, exploiting the attention between tags and the image features, opening the field of medical image captioning. Many others enhanced the performance achieved in medical image captioning by proposing various techniques like feature level attention(Wang [48]), using reinforcement learning(Li [57]), and using spatial attention over the localized image regions(Xiong [58]).

The success in medical image captioning has been possible due to the latest advances in deep learning. DenseNet [59], being a densely connected convolutional network, enabled us to learn high order dependencies by using a large number of layers with a minimal number of parameters, enabling the architectures to understand complex images like X-ray images without overfitting. Xception [22] proposed depth-wise separable convolutional operation, which in-turn extracts efficient image features with a decreased number of parameters in the model. Different training strategies like triplet loss function [26] and ranking based loss functions [24, 60, 61] also enhanced the performance of deep learning based systems for application problems. Moreover, the latest enhances in image captioning problems also played a vital role in developing radiology reports. Karpathy [62] performed image captioning using deep learning by providing the image features to the initial state

of RNN. The RNN then uses the state information to predict the caption of the image. Though RNN's capture temporal dependencies, they have substantial computational overhead. Transformers [27], on the other hand, can efficiently capture long and short term dependencies with minimal computation. Hence, this work tries to utilize the latest deep learning based techniques to generate accurate medical reports of radiology images.

4.2 Dataset used

Understanding the dataset is essential to appreciate the proposed methodology. For validating the proposed methodology, publicly available IU X-ray dataset [50] has been used. It contains the medical data of 3999 patients. Each data contains findings, impressions, MTI tags, and a set of N number of X-ray images taken for each patient. Findings and impressions are concatenated one after another to make the medical report of the patient. Medical text indexer (MTI) is used to extract keywords from the report forming the MTI tags (referred to as tags in this work). Since each patient has had multiple X-rays, there are a total of 7470 X-ray images. Each word of the report is tokenised and nonalphabetic tokens are removed. Moreover, for fair comparison, the frequency percentile of all the unique words in all the reports is computed and only the top 99 percentile of words are picked, which amounts to 1000. The dataset contains 573 unique MTI tags. Only those tags that appeared in at least three reports are considered for fair comparison with existing works. Hence we are left with only 283 tags for the tag prediction task. We discarded those patients' data, which did not contain either findings or impressions or X-ray images. For testing the performance of our proposed network, as suggested by Li [57], we randomly split patients for training/validation/testing in the ratio of 7/2/1.

4.3 Proposed Method

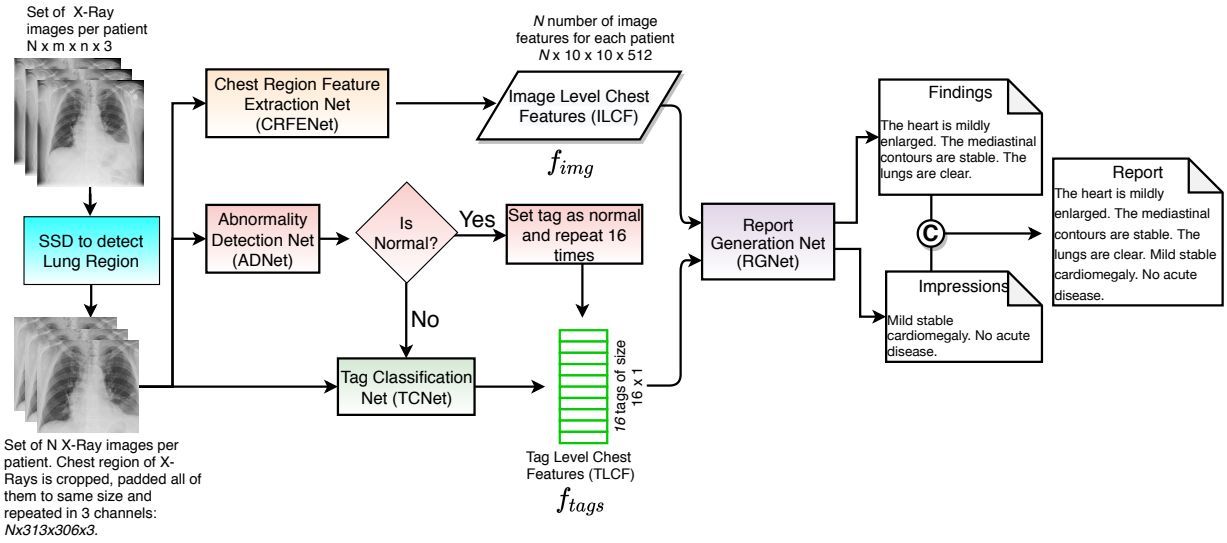


Figure 4.2: Shows the overall pipeline of the proposed system. The system's input is a set of X-Rays taken of a patient, and output is the generated medical report containing Findings and Impressions.

This section provides a technique that can generate accurate medical reports from a set of (N) X-ray images of one patient. Some of the images may cover the neck and abdomen portions too. To avoid the network from getting confused, first, a Single shot multibox object detector (SSD) [63] is trained to detect and crop the lung region from the given X-ray images. The images are then padded to a consistent size of 313×306. Figure 4.2 shows the overall pipeline that is proposed for generating medical reports from a patient's X-ray images. It consists of 4 modules, namely (i) Chest Region Feature Extraction Net (CRFENet), (ii) Abnormality Detection Net (ADNet), (iii) Tag Classification Net (TCNet), and (iv) Report Generation Net (RGNet). The CRFENet takes the input X-Ray image (I) of size 313 × 306 × 3 and provides a feature of size 10 × 10 × 512. This module is intended to provide contextual information of the image. ADNet also takes an input X-Ray image (I) and does a binary classification to identify any abnormality present. Since there is data imbalance with more data from healthy patients, a hierarchical classification

technique has been chosen to classify the samples between healthy and unhealthy classes, allowing conditional learning. Only the abnormal samples are put through the TCNet, which ranks the tags to their relevance to the report. Only the top sixteen tags are taken because the maximum number of tags associated with any patient is sixteen. In the case of a normal patient, all the sixteen tags are manually set to normal. Then, the RGNet takes image features and tags to generate Findings. Then in step 2, it takes Findings to generate Impressions. Finally, Findings and Impressions are concatenated to form the full report. For efficient training and hyper-parameterization, modular training and modular hyper-parameterization is employed. This section discusses each module, the training procedure, and the hyper-parameterization strategy in detail.

The significant contributions of this chapter are as follows:

1. Since in any consortium of diagnostic data a large number of normal patient data exists compared to abnormal patient data, a 2 stage divide-and-conquer approach is proposed. First, abnormal patients are identified from normal patients, and their tag embeddings are generated.

2. For predicting wordss in the report, a novel architecture has been proposed involving transformers with 2 encoders and 2 decoders each instead of traditionally used recurrent neural networks. This reduces training time and computational cost.

3. Tag embeddings and Image features are encoded separately using two different encoders. Findings and Impressions has different information and they can be learned by two stacked decoders, helping the former decoder to improve the generation of later.

This technique also surpasses the BLEU score (1-gram: 0.464, 2-gram: 0.301, 3-gram: 0.212, 4-gram: 0.158) of SOTA [47] by all the 4 n-gram metrics.

4.3.1 Chest Region Feature Extractor Net (CRFENet)

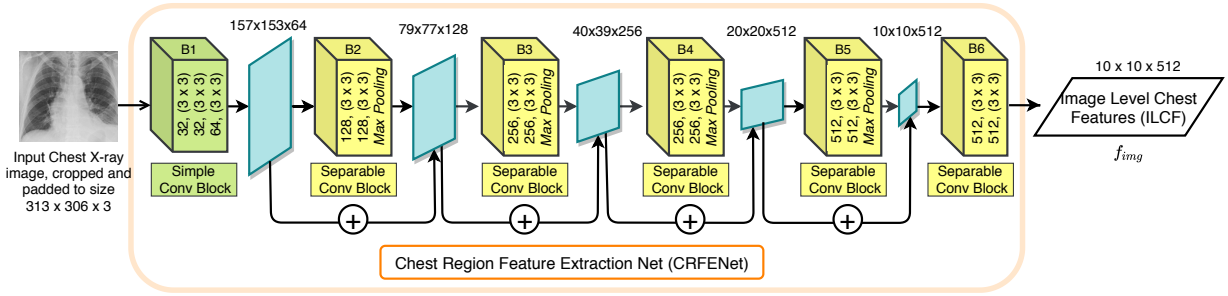


Figure 4.3: Shows the architecture of the proposed Chest Region feature extractor. The module contains residual blocks of depth-separable convolutions to decrease the number of overall parameters and computations. It helps to eliminate the over-fitting issues with medical datasets in which the available data is scarce.

The first task in generating automatic radiology reports is to identify the salient features present in X-Ray images that lead to the diagnosis. However, the challenge is that these features are complex to recognize and depends on type of analysis and they are highly non-linear. Hence, for extracting such features, complex non-linear function that can map an input image (I) to its feature ($fimg$)) has been learned as shown in Figure 4.3. A deep convolutional neural network (CNN) has been designed for extracting such sophisticated non-linear features. However, using deep CNNs have its own disadvantages, such as large number of parameters and vanishing gradient problems. Since medical datasets are scarce (this dataset has only around 3999 patient records), learning deep networks is difficult. The job is eased by incorporating the following two ideas in CRFENet :

- Use Depth wise separable convolutions [22] over simple convolutions in order to reduce the number of parameters.
- Reduced number of parameters also helps to deal with data scarcity.
- Use residual connections to solve the vanishing gradient problem of deep networks.

CRFENet contains one block of simple convolutional layers and four blocks of depth-wise

separable convolutional layers, as shown in figure 4.3. Batch-normalization and Relu non-linearity has been used after each conv layer to achieve better convergence.

Separable Convs: In convolution operation, the kernel aggregates the depth of input feature map to produce a single output. Hence, to generate an output having depth d, d such kernels are applied as already discussed in section 2.1. This will in turn produce vast amount of parameters that need to be optimized. Whereas in depth-wise separable convolution operation, one kernel is applied without aggregating depth-wise information. Hence, d pointwise convolution kernels are applied to provide us with the final feature with depth d as discussed in section 2.1. Using this technique, we can efficiently train a deep model with few parameters avoiding the overfitting problem.

Training: Learning a highly complex non-linear function to map image to its features is really a difficult task, especially when the data is scarce. For the CRFENet to understand how to observe the X-Ray images, first we trained it for chest disease classification on NIH Dataset [51]. The image features of IU Dataset extracted from CRFENet for final report generation are found to be better than the features extracted from deep CNNs like Dense121 [59] as shown in Table 4.1

4.3.2 Abnormality Detection Net (ADNet)

Conditional learning is done based on the status of the patient's data. We have to find whether a patient has any abnormality required to be reported in the generated report. To detect abnormal cases, we propose a binary classification module called ADNet. Classifying X-ray images into normal and abnormal classes. The patients who do not have any MTI tag associated with them are defined as normal patients. Figure 4.4 shows the detailed architecture of the proposed ADNet. As the abnormalities present in X-ray images are

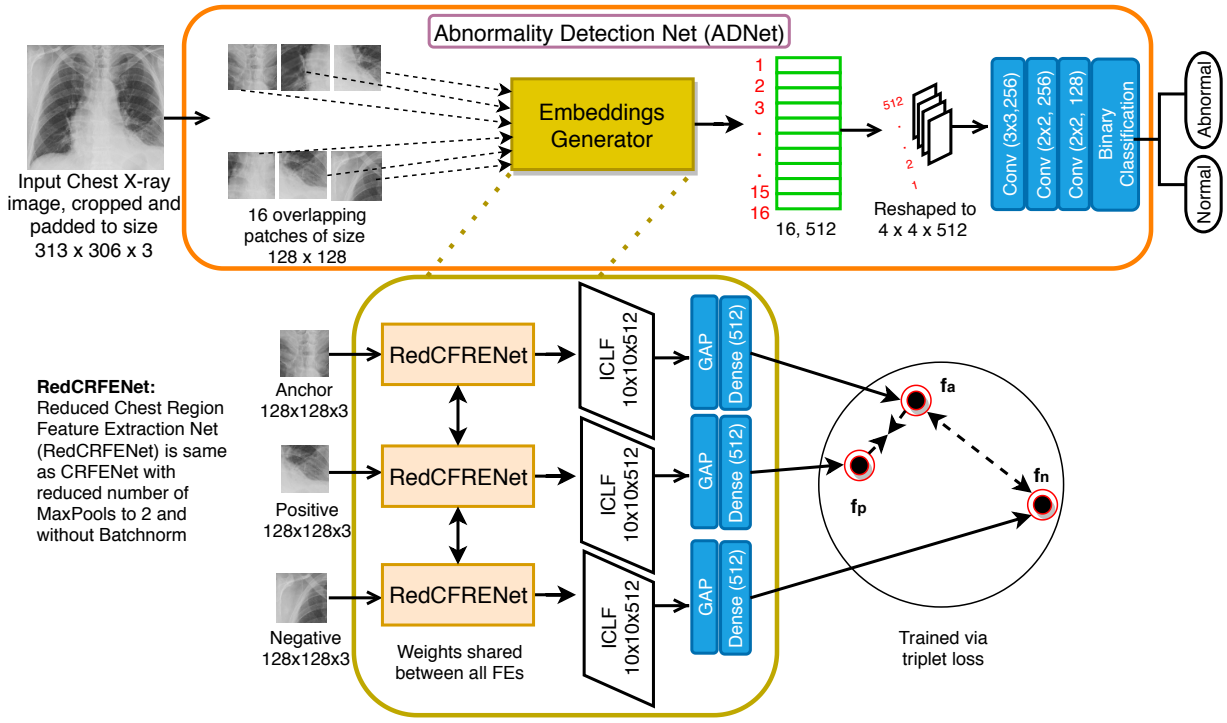


Figure 4.4: Shows the architecture of the Abnormality Detection Net (ADNet). It identifies the presence or absence of abnormality in an X-Ray image using the triplet loss function.

usually localized, it processes the images in patches. Each input image I is divided into 16 overlapping patches of size 128x128x3. These patches are then passed through a sub-network called Embeddings Generator (EG), which produces embedding of size (512x1) for each patch. The EG network is trained to produce embeddings such that normal patches and abnormal patches are as far as possible from each other in the proposed feature space. The 16 embeddings corresponding to a single X-ray image I are concatenated to form a feature vector of size (16x512) which is further reshaped to (4x4x512) to preserve the spatial relationship present between these patches. Intitially two Convolutional layers followed by a fully connected layer of 1 neuron has been applied to (4x4x512) feature vector to get the final classification probability. Since every patient's dataset contains a variable number (N) of X-Ray images, we take the average probability and threshold it at 0.5 to classify the patient as normal or abnormal.

4.3.2.1 Embeddings Generator (EG)

As discussed above, EG's task is to extract a 512-D feature from a patch (128x128x3) of the X-ray image. EG is trained via triplet loss function [26] inorder to learn a metric that can discriminate between normal and abnormal patches. rCRFENet contains two maxpool layers as compared to 4 in CRFENet. Each patch of size 128x128x3 is passed through rCRFENet, a reduced version of CRFENet, to produce the output of the same size 10x10x512 feature for each 128x128x3 patch. rCRFENet is pretrained on NIH data [51] because that contains the localization information of abnormality in X-Ray images, through an ROI. Patches of 128x128x3 are chosen around the ROI for training. Given two patches i and j, EG one patch at a time and produces an embedding Θ, such that if both i and j belong to the same class (normal or abnormal), then $L_2(\Theta^i, \Theta^j)$ should tend to 0, otherwise, $L_2(\Theta^i, \Theta^j) \geq \beta$, where β is the margin as discussed in section 2.2 in detail. The loss has been defined over 3 embeddings:

1. Θ^i: embedding of an anchor patch,
2. Θ^{i^+}: embedding of another patch from the same category, and
3. Θ^{i^-}: embedding of a patch from other categories.

Formally: $L(i, i^+, i^-) = \max(0, (\Theta^i - \Theta^{i^+})^2 - (\Theta^i - \Theta^{i^-})^2 + \beta)$; Loss for all possible triples (i, i^+, i^-) is summed to form the cost function J which is minimized during training of EG:

$$J = \frac{1}{N} \sum_{i=1}^{N} (i, i^+, i^-) \tag{4.3.1}$$

For efficiently training the EG network, online semi-hard negative mining and dynamic adaptive margin as proposed by [64] is applied as discussed in section 2.2.

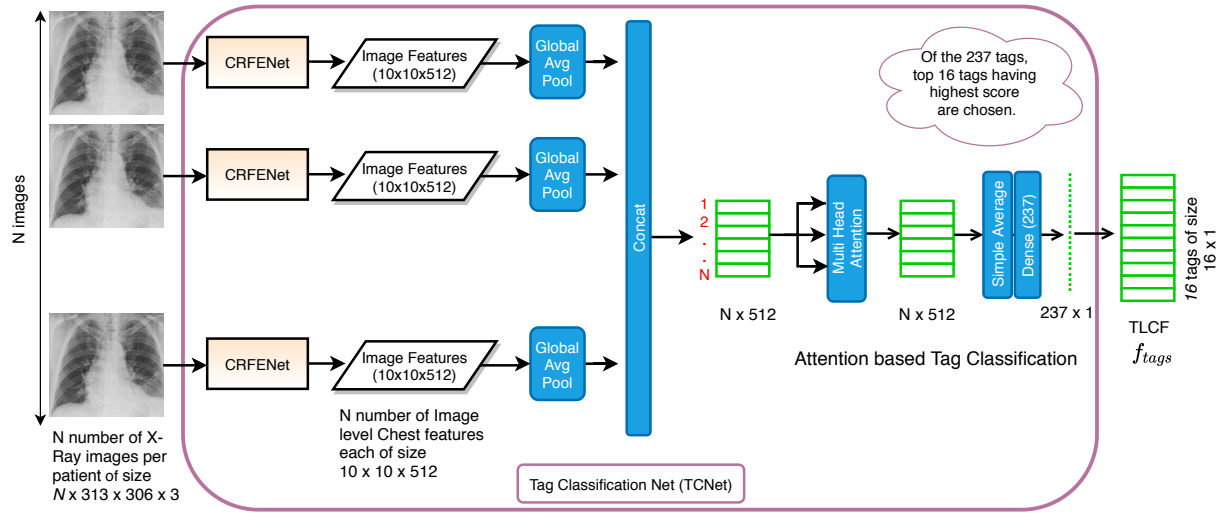


Figure 4.5: Shows the architecture of the proposed Tag Classification Net (TCNet). It generates the top 16 relevant tags about a set of X-Ray images of an abnormal patient.

4.3.3 Tag Classification Net (TCNet)

The second step of hierarchy, is tag prediction done by TCNet associated with each Pid_{img}. These MTI tags play a crucial role in generating the report. As shown in Figure 4.5, (N) images in (Pid_{img}) are passed through CRFENet, one after the other to obtain image features (10x10x512). Upon applying Global Average Pooling to each of the image features, we get (1x512) feature vector which is concatenated to get an (Nx512) feature vector. This (Nx512) sized feature vector passed through a Multi-Head Attention module (MHA) to get the same dimensional output. MHA computes attention across the (N) images and produces a result as discussed below. Over that, a simple averaging and Dense layer of 237 neurons is applied. TCNet is trained using log sum exponential pairwise loss function [24] as discussed in section 2.1.5. It assigns a value to each tag relative to other tags by learning to rank via pairwise comparisons. Finally, the values are sorted, and the top 16 tags are picked to produce an output (f_{tags}) of size (16x1).

Multi Head Attention (MHA) The basic building block of multi-head attention [27] is

the scaled dot product mechanism as discussed in section 2.4.4.3. The scaled dot product mechanism is a sequence to sequence operation: given a sequence of values vectors v_1, v_2, ..., v_n, it learns to provide an output sequence vectors y_1, y_2, ..., y_n, based on a query sequence q_1, q_2, ..., q_n, and key sequence k_1, k_2, ..., k_n, where each vector in the sequence is d-dimensional. First it learn three weight matrices of size $d \times d$ to transform each of the three sequences: $Q_i = W_q q_i \quad K_i = W_k k_i \quad V_i = W_v v_i$. Each y_i is computed as weighted average over the all transformed value vector V: $y_i = \sum_j^n w_{ij} V_j$, where j iterates over the whole sequence. Here w_{ij} is derived from dot product of query and key sequences: $w'_{ij} = \frac{Q_i^T K_j}{\sqrt{d}}, \quad w_{ij} = softmax(w'_{ij})$. Alternatively, in the scaled dot product mechanism, to compute one particular output y_i, the corresponding vector of query Q_i is compared (via. dot product) to the whole sequence of key vectors K_1, K_2, ..., K_n to provide the attention weights for each of the value vectors V_1, V_2, ..., V_n. Scaled dot product mechanism is used to form the multi-head attention mechanism. For a given set of value, query, and key vectors of $n \times d$, where n is the sequence length, and d is the dimensionality of each vector; Each vector is broken into r subparts of $n \times \frac{d}{r}$. We apply r different scaled dot product mechanisms, each having independent weight matrices of $\frac{d}{r} \times \frac{d}{r}$ giving us r outputs of $n \times \frac{d}{r}$. These outputs are concatenate to get the final output of shape $n \times d$. Here the total number of parameters is only $\frac{3d^2}{r}$ (3 weight matrices for each of r parts of the input sequence).

4.3.4 Report Generation Net (RGNet)

This network is based on transformer architecture inspired from [27]. RGNet consists of 2 Encoders called Data Encoder (E_D) and Tag Encoder (E_T), and 2 Decoders called Findings Decoder (D_{fin}) and Impressions Decoder (D_{imp}). The network architecture is shown in Figure 4.6.

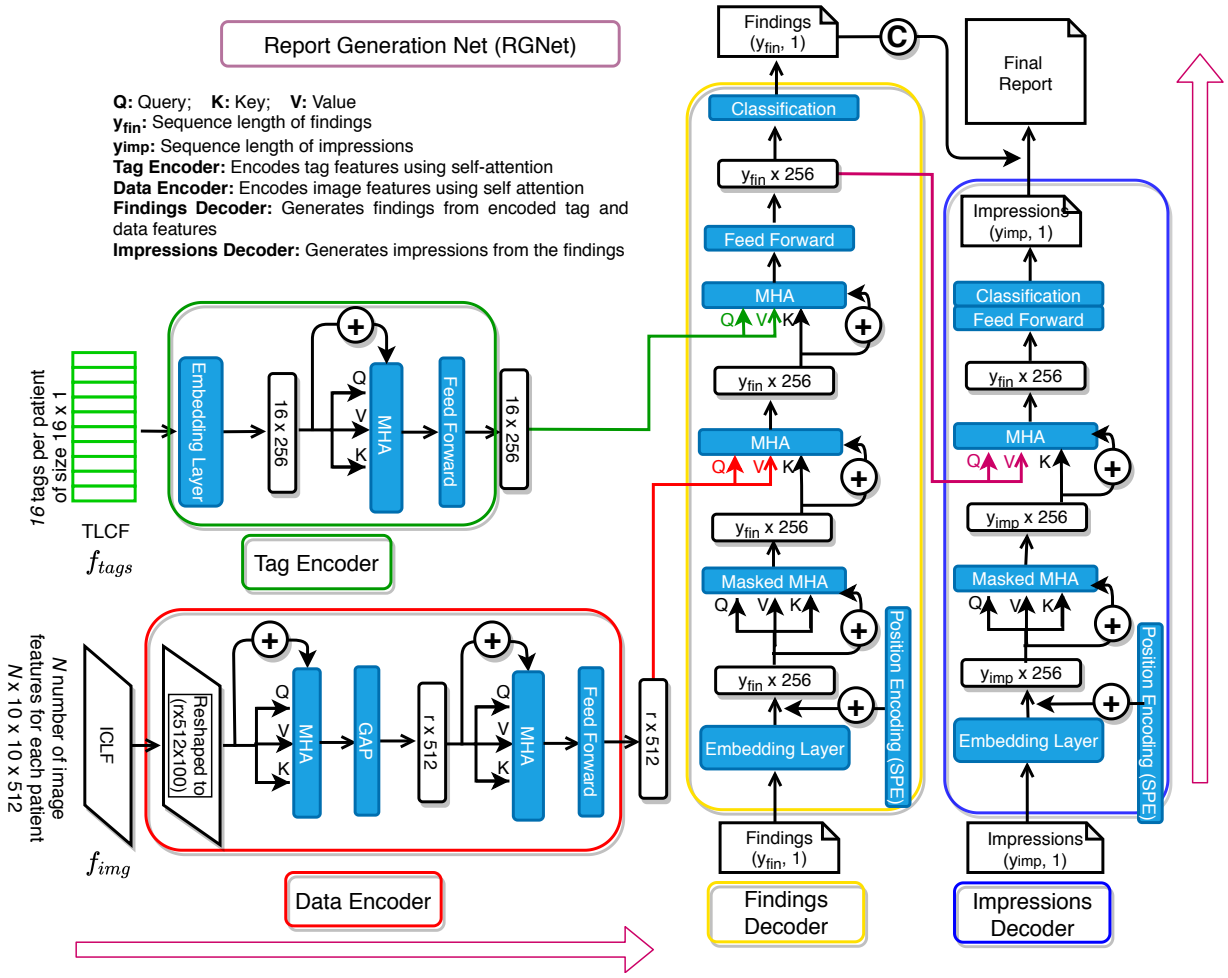


Figure 4.6: Shows the architecture of the proposed Report Generation Net (RGNet). This module generates the report using a blend of information from image feature and tag embeddings. Also, sequentially uses the report's Findings to generate the report's Impressions.

(I) Data Encoder (E_D): It takes N images of (Pid_{img}), passes one by one through CRFENet to get an output of size (10x10x512) for each image. The N features are then concatenated to form a feature of size (Nx10x10x512) and reshaped to (Nx512x100). Since neither all the N images are equally important nor every 512 features, we try to aggregate the appropriate features and images using 2 MHA modules. The first one learns self-attention over each of N image features providing us with a feature map of size (Nx512). The second MHA learns self-attention to combine the features across N images forming

feature embeddings efficiently.

(II) Tag Encoder (E_T): It takes the (16x1) tags extracted from TCNet and creates an embedding for each of the tags. Later an MHA is used to learn self-attention over the tag embeddings providing us with relevant tags only.

(III) Findings Decoder (D_{fin}): The task of (D_{fin}) is to generate the next word of Findings given a sequence of words corresponding to the tag embeddings, and image features. The next word will depend upon previous words as well as both tag embeddings and convolutional image features. A transformer block learns the attention required on previous words of the report over the tags embeddings and image features. Firstly, self-attention is learned on the previous words of the report. If words are indexed 0, 1,,, i-1, i, i+1 and so on, it consumes all the previous information (0 to i-1) to generate the next word(i). The future words (i+1 and further) are masked from the network. A multi-head attention mechanism is used to learn the self-attention, where the report is given as the key, query, and value. Secondly, cross-attention is learned between the output of the first self-attention block and image features. Multi-head attention mechanism is used again, but for learning cross-attention over tag embeddings. In both cases, the embeddings are given as value and key, where the previous attention block's output is given as query to multi-head attention block. Self-attention gives us the next word's dependence on previous words, whereas the cross-attention provides us with the dependence of the next word on the image features and tag embeddings. Both Self and Cross attention matrices update their parameters based on the loss generated w.r.t the next prediction. A feed-forward layer is applied after the cross attention forming the transformer block. An embedding layer is used to convert the words into embeddings of 256 dimensions, and then sinusoidal positional encoding (SPE) [27] is applied over the embeddings before inputting them into the transformer block. Finally, a linear layer followed by softmax cross-entropy loss gives

us the probability for each word in the dictionary to be the next word in the Findings.

(IV) Impressions Decoder (D_{imp}): Given a sequence of words corresponding to the Impressions and output feature from D_{fin}, (D_{imp}) generates the next word of the Impressions. It first learns the dependence of the next word on previous words by learning self-attention using MHA. Later cross-attention is learned between previous words of Impressions and the generated Findings from D_{fin}) using MHA, enabling the network to produce Impressions depending upon the previously produced Findings. Finally, the Findings and Impressions are concatenated to form the final report.

4.3.5 Training Procedure and Hyper-Parameterization

For training and searching for the optimal hyper-parameters of the proposed methodology, modular training approach has been used. We follow the below sequence of steps, and each model is hyper-parameterized for efficiently performing its pretraining task as discussed below:

- Pre-train the CRFENet and rCRFENet for chest disease classification on NIH Dataset [51].
- EG is trained using the triplet loss function over patches extracted from the NIH dataset to discrimate between normal and abnormal patches.
- ADNet is trained over IU-dataset for normal vs. abnormal classification.
- The pretrained CRFENet is used to finetune the TCNet using the ranking loss function in order to find the most relevant tags.
- Finally, RGNet is trained for report generation using the image features extracted from CRFENet and tags from TCNet.

4.4 Experimental Results

This section of the chapter provides details of the experimental analysis performed to validate the proposed methodology. Ablation study is performed for experimentally validating contributions proposed in this chapter. This experimental analysis reveals that the proposed model can produce accurate reports using qualitative and quantitative comparative analysis as discussed below.

4.4.1 Evaluation Metric

Captioning Evaluation Metric: For evaluating the report generated against the original report, standard image captioning evaluation metric Bilingual Evaluation Understudy (BLEU) score [65] is used. BLEU score measures the quality of the text generated and assigns a metric between 0 and 1. It analyses the statistics of overlapping words with the reference sequence. The original report is taken as a reference to run the string matching algorithm. A value of 0 means there is no overlap with the original report, and 1 signifies there perfect overlap with the original report. BLEU score compares n-grams of the candidate with the n-grams of the reference translation and count the number of matches. The 1-gram or unigram would be each token and a bigram comparison would be each word pair. These matches are position-independent. The more the matches, the better the candidate translation. Below is an example of its calculation:

Ground Truth/Reference (y): Heart size is within normal limits.

Predicted Output($\hat{y}$): The heart size is normal size.

- BLEU Score on Unigrams:

Unigrams $\in \hat{y}$	The	Heart	size	is	normal
Count$_y$	0	1	1	1	1
Count$_{\hat{y}}$	1	1	2	1	1

$$BLEU_1 = \sum_{unigrams \in \hat{y}} \frac{Count_y}{Count_{\hat{y}}} = \frac{4}{6} = 0.6 \tag{4.4.2}$$

Here, $Count_y$ is the number of times that particular unigram appears in the Ground truth y and $Count_{\hat{y}}$ is the number of times unigram appears in the prediction $\hat{y}$.

- BLEU Score on Bigrams:

Bigrams $\in \hat{y}$	The Heart	Heart size	size is	is normal	normal size
Count$_y$	0	1	1	0	0
Count$_{\hat{y}}$	1	1	1	1	1

$$BLEU_1 = \sum_{bigrams \in \hat{y}} \frac{Count_y}{Count_{\hat{y}}} = \frac{2}{6} = 0.3 \tag{4.4.3}$$

Here also, $Count_y$ is the number of times that particular bigram appears in the Ground truth y and $Count_{\hat{y}}$ is the number of times bigram appears in the prediction $\hat{y}$.

Similarly BLEU score can be defined for n-grams. It measures the quality of the text generated and assigns a metric between 0 and 1. It analyses the statistics of overlapping words with the reference sequence. The "Quality" is considered to be the correspondence between a machine's output and that of a human.

Classification metric: Classification performance provides accuracy percentage of correctly classified samples for a given class. It is defined as:

$$\text{Accuracy per class} = \frac{\text{Number of correctly classified samples}}{\text{Total number of samples available in the class}} \tag{4.4.4}$$

Task	Model	# layers	# parameters	Performance
FeatureExtractor	DenseNet	100	27.2 M	**0.175 Loss**
	CRFENet	**17**	**12.3 M**	0.19 Loss
AbnormalityDetection	DenseNet	100	27.2 M	70.5% Acc
	ADNet	**18**	**13.1 M**	**74% Acc**
TagClassification	TCNet with wBCE	19	12.6 M	0.44 Loss
	TCNet	**19**	**12.6 M**	**0.26 Loss**
Report Generation	**RGNet**	**12**	**0.7 M**	**0.464 Bleu-1**

Table 4.1: Parametric comparison and modular ablation analysis. Acc: Accuracy, wBCE: weighted Binary Cross Entropy, M: Million. Bold represents the proposed systems.

Model	Bleu-1	Bleu-2	Bleu-3	Bleu-4
Model A(without CRFENet)	0.414	0.287	0.198	0.143
Model B(without ADNet)	0.320	0.218	0.156	0.116
Model C(without ranking loss)	0.295	0.192	0.104	0.092
Model D(without 2 decoder RGNet)	0.423	0.292	0.204	0.148
Proposed Methodology	**0.464**	**0.301**	**0.212**	**0.158**

Table 4.2: Ablation study of the proposed methodology validating our contributions.

4.4.2 Ablation Study

In order to validate the contributions of the proposed network, an extensive ablation study has been performed, as shown in Table 4.2 and 4.1. It is important to note, though the system is broken into multiple modules, the complexity of the overall system (38.7 M parameters and 66 layers) is comparable or lesser than state-of-the-art systems. Table 4.1 shows the parametric comparison and performance of each of the individual modules concerning corresponding state-of-the-art systems. Proposed methodology is shown in the first row of the Table 4.2 containing four modules named CRFENet, ADNet, TCNet, and RGNet, as described in Section 3. Ablation study has been demonstrated with alternatives for every network mentioned above to testify each of the proposed networks' ability and its contributions in the following order:

- Firstly, The system has been tested by replacing the proposed CRFENet with pre-trained state-of-the-art CNN architectures (Model A). Among such architectures,

Densenet[59] provided the best performance, as shown in the 1st row of Table 4.2. Since CRFENet is only a six-block module with separable convs, fewer parameters prevent over-fitting.

- Secondly, instead of ADNet, a simple VGG network is used for abnormality detection (Model B). Since most of the X-rays' abnormalities are localized, patch-based siamese abnormality detection (ADNet) provides us better results than standard-sized image-based classifiers like VGG.

- Thirdly, TCNet is trained with weighted binary cross-entropy loss rather than ranking loss (Model C). Since most of the tags are only associated with very few reports, the ranking loss is better able to capture the association of tags with particular X-Ray images.

- Finally, single decoder RGNet is trained, rather than the proposed two decoder RGNet (i.e Model D). It has been observed that The sequentially trained stacked decoders', one for Findings and Impressions can learn better to optimize their respective models. It will force the network to generate accurate Findings so that better Impressions can be generated and vice-versa.

It is evident from Table 4.2, that each of the ablated models performs inferior to the proposed network, which concretely validates each modules' contribution is significant and important. We can also notice the magnitude of the gain obtained from each of these changes. Without ranking loss Bleu-1 score for report generation is 0.169 units less than the proposed methodology. Ranking loss and hierarchical tag classification techniques gave the biggest improvements in the report generation's quality.

Model	Bleu-1	Bleu-2	bleu-3	Bleu-4
S&T [66]	0.265	0.157	0.105	0.073
SA&T [67]	0.328	0.195	0.123	0.080
TieNet [48]	0.330	0.194	0.124	0.081
Lie [68]	0.359	0.237	0.164	0.113
CNN-RNN [55]	0.216	0.124	0.087	0.066
LRCN [69]	0.223	0.128	0.089	0.067
AdaAtt [70]	0.220	0.127	0.089	0.068
Att2in [71]	0.224	0.129	0.089	0.068
RTMIC [58]	0.350	0.234	0.143	0.096
Li [57]	0.438	0.298	0.208	0.151
CoAtt [47]	0.455	0.288	0.205	0.154
Proposed Methodology	**0.464**	**0.301**	**0.212**	**0.158**

Table 4.3: Comparative analysis of the proposed system with state-of-the-art.

4.4.3 Quantitative Comparison

Table 4.3 shows the comparative analysis of the proposed system with state-of-the-art networks. It can be seen from the table that our proposed methodology achieves state-of-the-art for report generation task. Key components of our proposed methodology like hierarchical tag classification, ranking based loss, attention-based feature extraction, and transformer architecture are the leading cause for our model's performance to be better than the rest as also suggested in Ablation study.

4.4.4 Qualitative Comparison

Figure 4.7 and Figure 4.8 shows the qualitative results of the report generated from our proposed network. Figure 4.7 depicts examples from high accuracy outputs, whereas the last two rows contain the failure cases. In the first row, the proposed system can correctly identify "calcified granulomas" and generate a technically sound report. In the second row, the proposed system identifies "cardiomelagy". Moreover, for cases where there were only degenerative changes, our method performed well. We can also observe from the

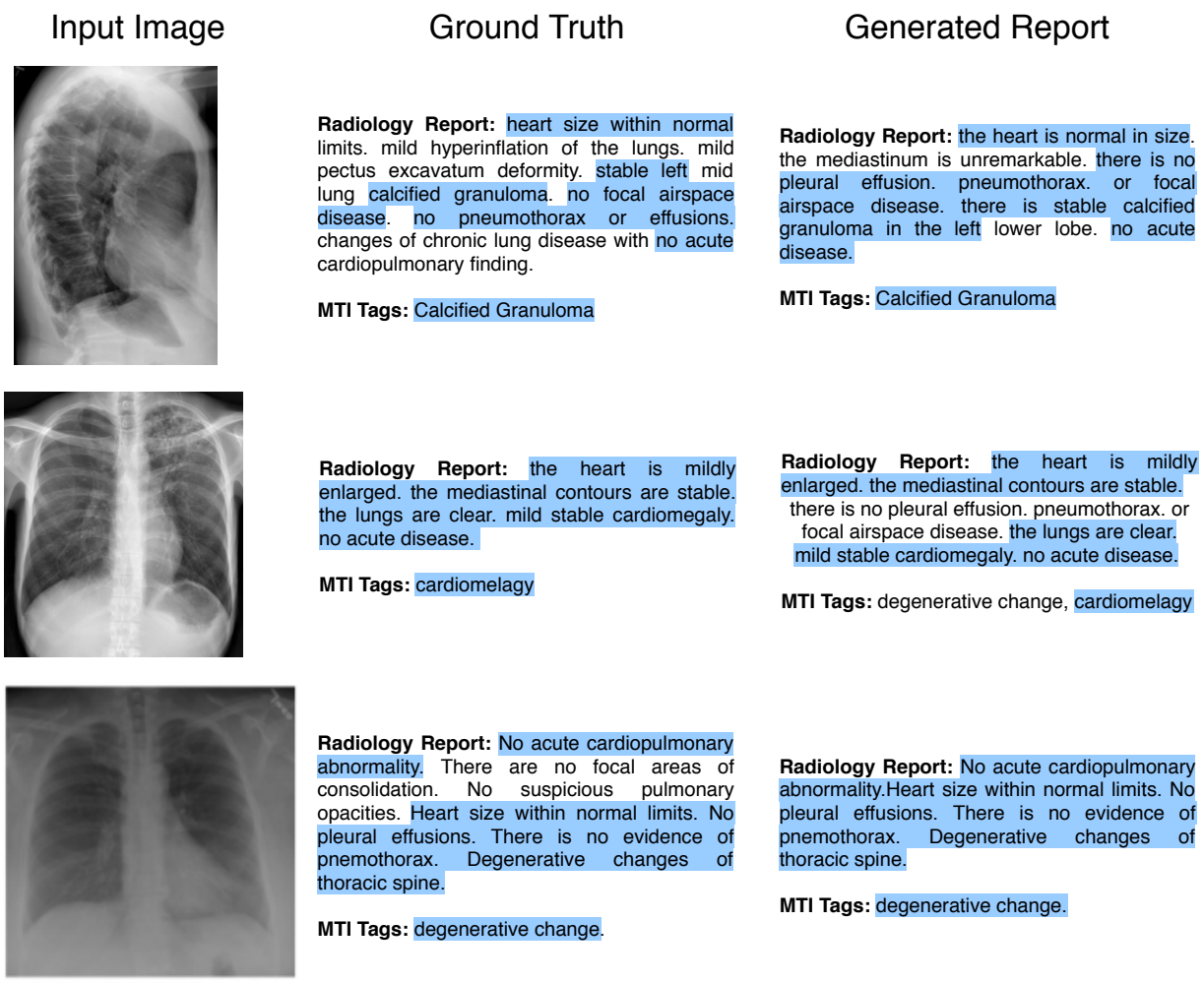


Figure 4.7: Shows the qualitative results of report generated from our proposed network. The rows depict examples from high accuracy outputs. The correctly predicted vocabularies are highlighted.

highlighted portion that most of the report's predicted characteristics match the original report.

Two significant cases of failures were found and one example of both is depicted in the Figure 4.8. The first case being rare abnormalities that only come ones or twice in the dataset. In such cases, the proposed system was not able to learn about them. The second case of failure is where the images were blurry or hazy. In such cases, the network predicted the patient to have no disease at all.

Input Image	Ground Truth	Generated Report
	**Radiology Report:** status post midline sternotomy with intact. stable mild cardiomegaly. normal lung vascularity. the lungs are clear. stable postop changes with stable mild cardiomegaly and normal lung vascularity. **MTI Tags:** sterntomy	**Radiology Report:** the heart is normal in size. the mediastinum is unremarkable. mild pectus excavatum deformity is noted. the lungs are clear. no acute disease. **MTI Tags:** pectus excravectum.
	**Radiology Report:** cardiomediastinal silhouette is unchanged with mild cardiomegaly. there is relative elevation of the right hemidiaphragm consistent with history of right lower lobectomy. without focal consolidation. pneumothorax. or effusion identified. irregularity of the right <unk> and <unk> ribs stable since at <unk> and <unk> postsurgical <alt> post traumatic in <unk>. left shoulder rotator <unk> bone <unk> noted. **MTI Tags:** cardiomelagy, lebactomy	**Radiology Report:** No acute cardiopulmonary abnormality. There are no focal areas of consolidation. No suspicious pulmonary opacities. Heart size within normal limits. No pleural effusions. There is no evidence of pnemothorax. Degenerative changes of thoracic spine. **MTI Tags:** degenerative change.

Figure 4.8: Shows the qualitative results of report generated from our proposed network. The rows depict examples of cases where the prediction is not accurate.

4.5 Conclusion

Captioning medical images is a complex task because, unlike the natural images, the salient features are not apparent. In this Chapter, a technique is proposed to blend the image and tag features and use it in a unique way to generate a medical report from a patient's set of X-Ray images. Traditional use of recurrent neural networks (RNNs) to solve such sequential data has a massive sequential computational overload. On the other hand, transformer architecture, which also captures the attention and relevant temporal dependencies from sequential data, uses far fewer parameters. Furthermore, it applies attention between and across features obtained from images, tags, and reports. While significant improvements have been achieved over the SOTA, there is still scope for improvement in generating useful quality reports, especially from hazy X-Rays or cases where different X-Rays are acquired under different exposures.

Chapter 5

Medical Image Denoising

Medical images are prone to noise like any other natural image. The presence of noise can make the images unclear and hard to identify and diagnose diseases which can result in fatalities. Therefore, medical Image denoising can play an important role in processing steps for further investigations. Some of the key considerations of a denoising algorithm can be - (i) Flat areas such as liver region as shown in the red circle of Figure 5.1(a), may become smooth as in Figure 5.1(b), (ii) Edges need to be as sharp and clear as possible. As shown in the green circle of Figure 5.1(a), edges of the pancreas region are blurred and smudged into the neighbouring organs. While after denoising as in Figure 5.1(b) edges of the pancreas became distinct, (iii) Textures must be preserved, and (iv) New artefacts must not be generated. Denoising neither must remove teh select features that exist nor can it introduce what does not already exist. Its goal needs to be to decrease the noise while preserving the original features and improving the signal-to-noise ratio. This chapter aims to propose a novel technique to generate denoised images which can help the diagnosis.

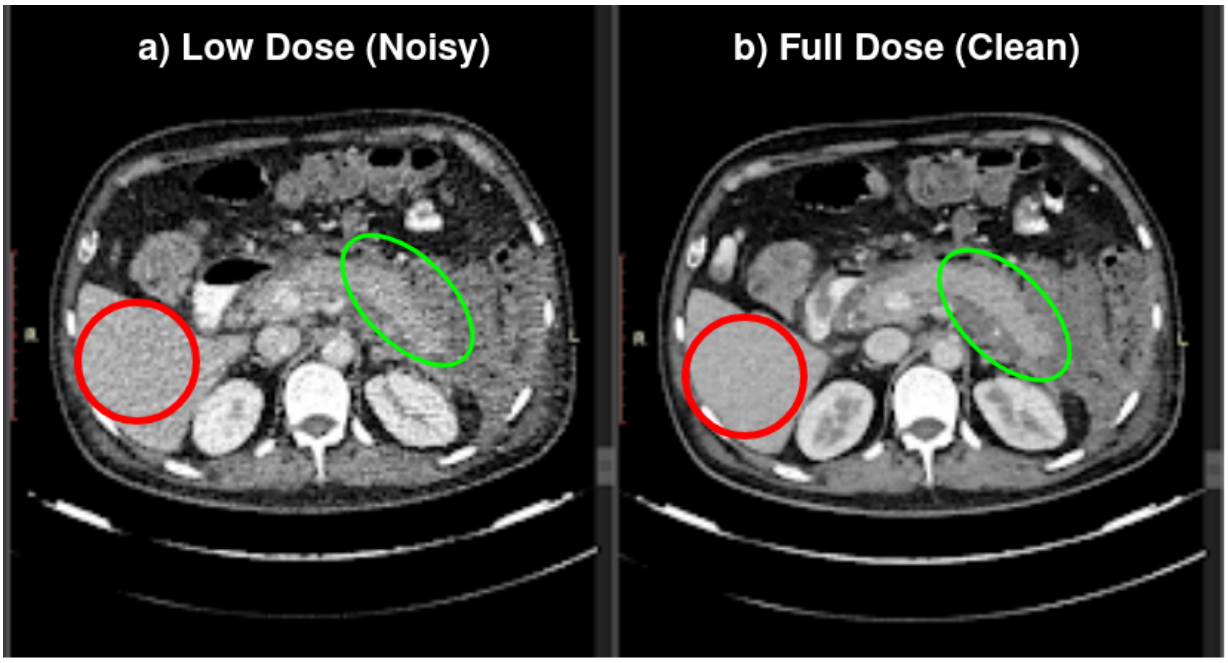


Figure 5.1: CT Image: (a) 70kV Low-dose has more noisy grains compared to the (b) 100kV Full-dose scan. This slice is from an anonymous patient in the dataset provided by PGI Chandigarh hospital. Red circle is on the liver region which is a large flat area and Green circle is on the pancreas region which relatively less flat.

5.1 Denoising Low-Dose CT Scan

Computed tomography (CT) is a non-surgical procedure used to visualize 3-dimensional internal structures of organs at a high resolution. As shown in Figure 5.2(1) CT machine rotates and takes a series of X-ray images from different angles called the projected images. These projected images are combined by eliminating the overlapping areas to create 2D slices across the cross-section of the anatomy, making a 3D volume as shown in Figure 5.2(3a and 3b). Therefore a tomographic reconstruction is defined as estimating the original object from its projection. The thickness of a 2D slice varies from 1mm to 10mm. For a given patient, the 3D volume required to cover the abdomen region is fixed and as the thickness of the slice increases, the number of slices necessary to form the 3D volume becomes fewer. Thinner slices contain more information than thicker slices. A CT Scan is well suited for

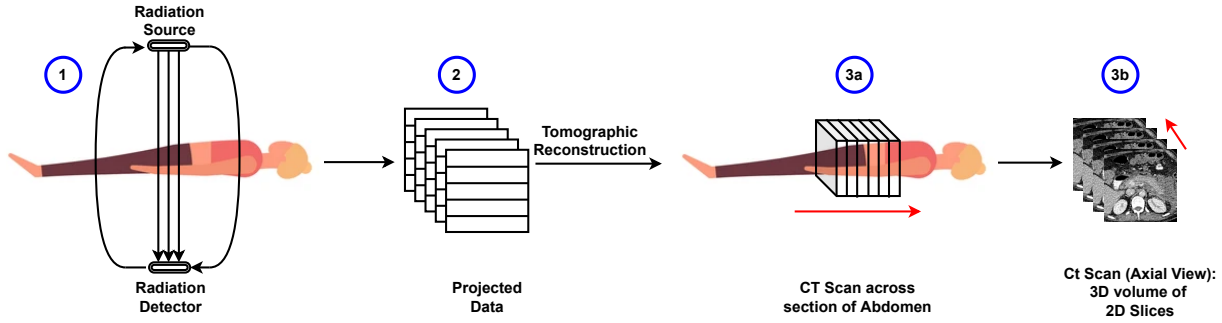


Figure 5.2: Acquiring process of a CT Scan: (1) X-ray radiation flows through the body from the source to detector while rotating. (2) X-rays produce projected data. (3a and 3b) Tomographic reconstruction is applied on projected data to estimate the 2D slices comprising the 3D volume of CT Scan. Red arrow shows the point of view (Axial).

detecting cancer, pneumonia, abnormal chest issues and bleeding in the brain after an injury. For example, it plays an integral role in diagnosing Acute Pancreatitis which is an infection caused due to cells dying in the Pancreas. This may lead to an infection in Pancreas and may spread to the neighbouring organs. CT is also used to monitor the spread of the infection and evaluate fluid collections in the damaged area.

CT involves two types of radiation exposure depending on the amount of X-ray radiation passed through the body: Low dosage of radiation produces noisy images as compared to the images produced by high dosage of radiation (as shown in Figure 5.1). Tube current(mA) and Voltage(kV) are the two machine parameters that effects the CT's radiation dosage. Decreasing the magnitude of these parameters reduces the amount of radiation but the images produced are very noisy and hard to understand. For example, keeping all the machine attributes, including the tube current(mA) constant, a 70kV scan will be noisier than a 100kV scan. Vice versa keeping the voltage(kV) constant, lower current will induce more noise in the image. In literature[72] the statistical distribution of noise in CT images is regarded as a mixture of Poisson quantum noise and Gaussian electronic noise.

High radiation exposure contributes to a substantial risk of an increased number of

cancer patients in the population. While repeated high dosage CT Scans might be necessary for accurate assessment, exposure to high radiation multiple times makes the patient more susceptible to cancer, especially in the case of the elderly population and children[73, 74]. The protocol for dosage levels followed in the hospitals is usually ALRP (As Low as Reasonably Possible). However, such flexibility in protocols may not be practical during implementation.

5.1.1 Motivation and Problem Statement

MRI can produce a higher resolution image than CT Scan. However, it takes 45 minutes while CT Scan takes only 10 minutes. Avoiding MRI can increase comfort due to the lower acquisition time of CT Scans. In addition to that typically, CT Scan costs much less than a MRI Scan. For these reasons, CT Scan is preferred during emergency conditions. However, repeated high dose CT Scan can increase the risk of cancer. Therefore, denoising the low dose CT (LDCT) to appear as good as a full dose CT (FDCT) can be a important practical problem to solve.

Furthermore, to learn the transformation function from LDCT to FDCT during training, one may typically use the corresponding pairs of LDCT and FDCT from the same patient. The data setting where both LDCT and FDCT are available from the same patient is called Paired data, and the training paradigm on paired data is called Supervised training. Though, it's not practical to ask the patient to take multiple scans under different radiation doses for research purposes. Hence we have conducted experiments for three different training scenarios based on the percentage of paired data.

- Case 1: Supervised training on paired data - 100% paired data, which means each pair of LDCT and FDCT is acquired from the same patient.

- Case 2: Un-supervised training on unpaired data - 0% paired data, which means,

no pair of LDCT and FDCT is acquired from the same patient. All the scans are acquired from totally different patients.

- Case 3: Semi-supervised training on semi-paired data - Only a few pairs of LDCT and FDCT Scans are taken from the same patient, and the rest of the LDCT and FDCT Scans are from different patients.

Interestingly, task of denoising addressed in this chapter, can be considered as an image transformation, another example of which is also addressed in chapter 3. Since the proposed method in Chapter 3 is broadly a Unet based architecture, we considered the application of Unet for the denoising problem as well. However, we noticed a significant performance improvement when (MHA) Multi Head-Attention modules, as employed in Chapter 4, is adapted for image feature extraction and image reconstruction for this case. Thus, we propose a DNN architecture that can generate denoised CT Scans using patch-wise multi-head visual attention blocks and residual connections to denoise Low dose CT Scans. The proposed network has obtained a significantly higher PSNR and better quality reconstruction over the State of the art Architectures - SACNN[9] and Unet[5].

5.1.2 Related Works

In the literature of denoising most of the solutions broadly fall into three buckets.

- Sinogram filtering smoothens the raw images itself (Filtered Back Projection - FBP). Some other methods in this domain include Bilateral filtering [75], Structural adaptive filtering [76], Penalized weighted least-squares algorithms [77], etc. They work directly on the projected data at the image preprocessing stage. However, they are difficult to practise because raw data (projection data) is not easily available.

- Iterative Reconstruction (IR) algorithms incrementally estimate the enhanced images using prior information about the image. These also work directly on the projected

data at the image preprocessing stage. Some methods use Total Variation which states that images with spurious details have high total variation and minimizing it can remove the noise [78, 79, 80, 81]. Similarly, few other image priors such as Non-Local Means have been utilised. Unlike local mean filtering, which takes into account only the neighbouring pixels around the target pixel, non-local mean filtering uses all the pixels of the image [82, 83, 84]. Some reported methods use Dictionary Learning [85, 86]. Although these techniques are very effective, they are computationally expensive and take a long time to reconstruct the denoised images.

- Post-processing algorithms work on the reconstructed images instead of directly working on the projection data. Classical methods like Non-local means filtering [87, 88, 89], Dictionary Learning based algorithms [90, 91] and block matching techniques [92, 93, 94] have been utilized. These perform better than IR techniques in terms of computation efficiency. However, the performance of qualitative reconstruction is not desirable because the transformation function to be learned is highly non-linear for them. The revolution of Deep Learning (DL) methods has given rise to multi-layered networks that can approximate such mapping functions from LDCT to FDCT, that are highly complex. The DL methods can effectively formulate practical noise scenarios. Some of the recent deep learning-based architectures to address such issues are discussed in Table 5.1.

5.2 Dataset Used

Inorder to justify our proposal we have utilized two datasets. **Dataset I:** The performance of the proposed architecture over the existing solutions, we have used the publicly available dataset that is released as a part of the 2016 NIH-AAPM-Mayo Clinic Low Dose CT

Type of Solutions	**Deep Learning based works**
2D Convolution based solutions	[95] proposed a encoder decoder network with residual connections, [96] proposed to use cascasded CNN instead of residual connections, [97] proposed a solution where along with low dose image, coutourlet transform of 4 levels leading to 15 channels are concatenated and sent to the network, and [98] proposed a 3 layered convolutional network mimicking the 3 steps of iterative reconstruction.
3D Conv based solutions	[99] compares the perceptual features of a denoised output against those of the ground truth in feature space to keep the critical structural information intact, [100] proposed a network with dilated convolutions and residual connections which trains on MSE loss along with perceptual loss, [101] proposed a 3D convolutional network based WGAN which improves on simple WGAN and [102] used the idea of transfer learning technique to copy the set of trained 2D conv filters inside the 3D conv filters and to make the rest of the filters 0 before starting to train the network.
GAN based solutions	[103] compared and showed WGAN works better than a simple CNN, [104] proposed a simple GAN based solution, [105] proposed a conditional GAN trained on adversarial loss and sharpness loss and [106] proposed a conditional GAN trained on adversarial loss and sharpness loss
Attention based solution	[9] proposed a attention based solution. Applies attention within the slice and across slices of certain depth. Computes perceptual loss using customized Encoder-Decoder network instead of VGG and adverserial loss using WGAN.
Super-resolution based solution	[107] proposed super resolution based 3D convolution to process the images.
Semi-supervised solutions	[108] [109] proposed a semi supervised learning method for cases where some data is jumbled and no corresponding data is available

Table 5.1: Deep Learning based related work

(LDCT) Grand Challenge [1]. The challenge contains Abdomen CT Scans of 10 patients. Later they all got populated are made publicly available at National Cancer Institute's The Cancer Imaging Archive (TCIA) [2] and the details of the data is available at Mayo.edu library [3]. TCIA contains data of 50 subjects with corresponding pairs of LDCT and FDCT. All the Abdomen CT Scans have been acquired at routine dose levels using standard clinical protocols. Later, Poisson noise has been inserted into each projection dataset to create a second projection dataset simulating lower dose level. The scans are provided at 25% of the routine dose. The simulation technique accurately models the impact of tube current modulation, the bow-tie filter, and electronic noise[110]. We have used LDCT images as input, and quarter dosed FDCT images as the target image. Each scan is a 3D volume having a variable number of 2D slices ranging from 100 to 250. The pixel intensity of each slice varies from 0 to approximately 2000.

Dataset II: To prove the efficacy of our network on real world data, we have used a clinical dataset of Abdomen scans, acquired at the Gastroenterology department of PGIMER, Chandigarh [4]. As discussed in section 5.1, 70kV data is low dose CT, and 100kV data is the full dose CT in this dataset. Each 3D volume comprises a variable number of 2D slices ranging from 200 to 300 with a thickness of 1mm. Six scans (2108 slices) are used for training, two scans are used for validation, and two scans (510 slices) are used for testing.

5.3 Proposed Method

This section aims to propose a novel deep learning architecture that can denoise a LDCT (I_{LD}) to appear like a FDCT (I_{FD}) for better clinical diagnosis. As shown in Figure 5.3,

[1] https://www.aapm.org/grandchallenge/lowdosect/
[2] https://public.cancerimagingarchive.net/nbia-search/
[3] https://ctcicblog.mayo.edu/hubcap/patient-ct-projection-data-library/
[4] https://pgimer.edu.in

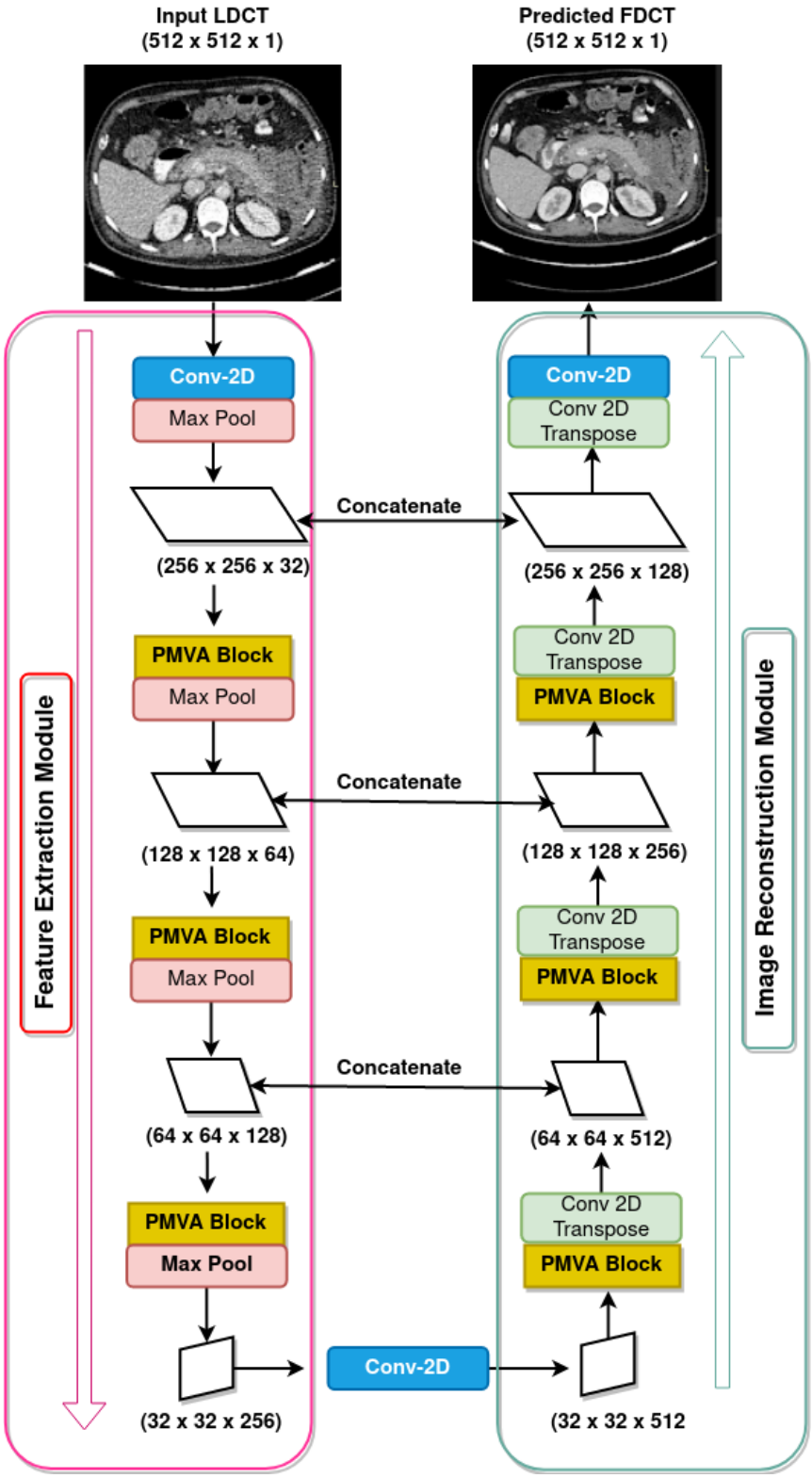


Figure 5.3: The overall architecture of PMVA-Unet.

the overall architecture has two modules - the Feature extraction (FE) module and the Image reconstruction (IR) module, and it is called as PMVA-Unet: Patch-wise Multi-head Visually Attentive Unet (denoted as ψ in the rest of the chapter). As opposed to using convolutional layers for feature extraction, as in Unet and most other popular image classification networks, we applied multiple blocks (PMVA block) of multi-head visual attention across non-overlapping patches to extract features in both the modules. The PMVA block is followed by the Maxpool layer (explained in Chapter 2) in the FE module and the Conv-Transpose layer in IR module. Conv-Transpose operation extrapolates the input into a bigger size using learned features. The residual connections between the blocks of both the modules preserve the information during reconstruction. The PMVA-Unet (ψ) can be used when data is 100% paired (Case 1 - Supervised training). We use L_2 distance (MSE loss) to guide the training process.

$$L_2(\psi) = \frac{1}{M}\sum_{i=1}^{M}[\psi(I^i_{LD}) - I^i_{FD}]^2 \tag{5.3.1}$$

where $\psi(L^i_{LD})$ denotes the predicted output from the proposed network PMVA-Unet. Its expectation over all M samples in a batch are computed and its derivative is backpropagated through the network to optimize the parameters and decrease the loss in the following iterations.

The contributions of this chapter are as follows:

- CT image $(I_{LD}, I_{FD}) \in \mathrm{R}^{512*512*1}$ is very large compared to the image sizes which many computer vision algorithms deal with. So, most operations over this size will require an exponential space and time cost. To the best of our knowledge, all the works done to solve this problem using DL so far have used small pieces of input image to the network (for example, $64 * 64$ in case of SACNN[9]). However, the network cannot keep global information in context, while reconstruction, if the input is processed in pieces. In this work, we process the entire image as input to the network but apply

operations across non-overlapping patches to balance both - having global context while adhering to memory limitations.

- Convolution operation operates only on a specific region of pixels at a time and thus is inefficient in capturing global information from the whole CT Scan. On the other hand, attention finds relevance of each input token with every other token. Therefore, transformer blocks[27] which were originally made for the language translation task, has been adapted for feature extraction from images by replacing the Feed-forward layer (MLP: Multilayer Perceptron) with the Convolution layer after MHA (Multi-Head Attention) block.

- In addition to the supervised training with paired data scenario, we have proposed techniques for semi-paired (Case 2) and unpaired settings (Case 3) of data availability.

5.3.1 PMVA Block

Our first two contributions are together performed in the PMVA block (Patch-wise Multi-head Visual Attention block) in two different modules:

- **Patch-wise Embedding Generator (PEG) module:** Several internal organs such as Liver, Pancreas, Kidneys, tissues etc, can be observed in a Abdomen CT Scan. Each of these regions has different smoothness levels. As shown in Figure 5.1, the Liver is a lot smoother than other organs. So, attention across non-overlapping patches finds relevance between the smoothness of different regions and improves the denoising procedure. Since attention is computed across all the patches, it simulates non-local means filtering by considering neighbourhood patches to smooth a given patch. For a given input feature map $x \in R^{H*W*C}$, where H denotes height, W denotes width, and C denotes the number of channels, the input is reshaped into $N * C$ where $N = H * W$ by flattening the pixels of each feature map. As shown

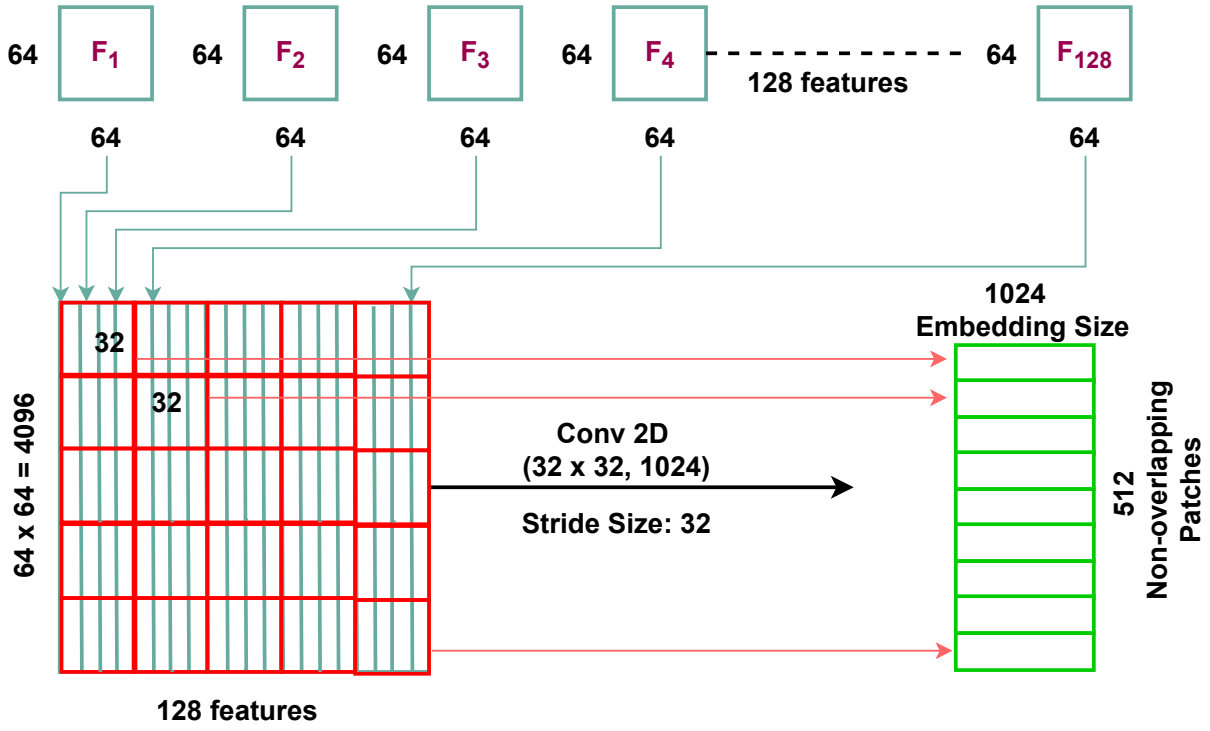


Figure 5.4: Workflow of Patch-wise Embeddings Generator (PEG) module. Feature output of size $64 * 64 * 128$ is reshaped into $4096 * 128$ and a conv layer with 1024 filters, each of size $32 * 32$ with stride size 32 is applied to get an output of size $512 * 1024$. Each non overlapping patch (shown in red color) is mapped in the embedding space to a vector of size $1 * 1024$.

in Figure 5.4, a conv filter of size $32 * 32$ is applied over the input with a stride of size 32 producing an output of $PQ * 1$ where $P = N/32$ and $Q = C/32$, which is equal to the number of non-overlapping patches in the input. We have applied 1024 such conv filters to produce an output of size $PQ * 1024$. This can also be seen as each of PQ non-overlapping patches of size $32 * 32$ are converted into an embedding vector of size $1 * 1024$ using a conv layer. The output from here is further fed into the Multi-head Visual Attention module.

- **Multi-head Visual Attention (MVA) Module:** A feed-forward layer (FFL) is used in Transformer model architecture[27] following MHA (Multi-head attention) for introducing non-linearity in the overall function. There are two disadvantages of

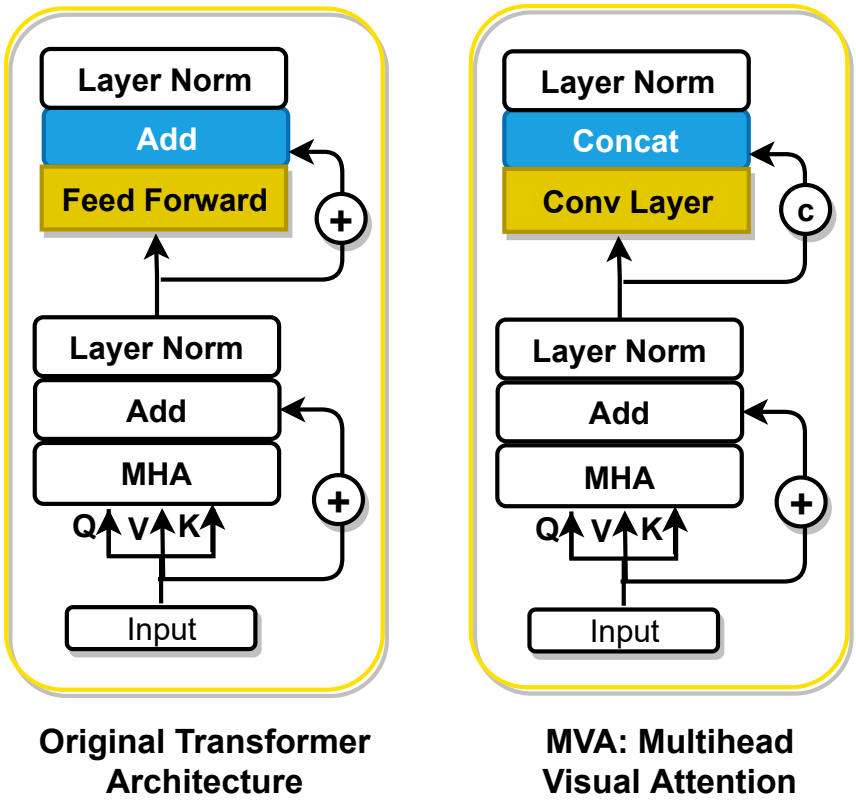


Figure 5.5: Difference between Original transformer architecture for language translation and MVA for image feature extraction

using FFL while processing images, especially when the size is large, as in the case of a CT Scan image. (i) Amount of computation is very high because MLP is full matrix multiplication, and vectors of size $4096 * 128$ create out-of-memory issues. (ii) FFL does not exploit the spatial information as compared to a convolution operation can. Hence, we have replaced the FF layer with a conv layer in the Transformer block as shown in Figure 5.5. While in the case of the original Transformer architecture, the features from MHA are added to the output of FFL, here features out of MHA are concatenated with the output of the conv layer. Concatenation avoids the mixing of information so that the network can choose the useful features required at the next step. MHA is explained in detail in section 4.3.3. It takes an input of size $PQ * 1024$ and returns an output of the same size $PQ * 1024$ after computing attention across

all the patches.

5.3.2 Un-supervised training on unpaired data:

In this section, we have worked on unpaired LDCT and FDCT data. The dataset-I has 10 subjects scans out of which 6 scans are used for training. We have used 2 LDCT scans and 4 FDCT scans from 6 different patients under this un-paired set up. We have not considered paired data in this scenario. The network is expected to learn mapping function without any specific corresponding LDCT and FDCT slices. In each epoch, a batch of LDCT has been given as an input, and a batch of FDCT images from different patients have been used as the target for training. As discussed in Section 2.5, CycleGAN [1] is a training paradigm that uses GAN architecture to address the problem of learning mapping function from an unpaired collection of data of one domain to another without one-to-one corresponding. Hence we have utilized CycleGAN's training paradigm utilizing cyclic consistency loss, identity loss and adversarial loss.

As shown in Figure 5.6 ψ_{L2F} learns mapping function from $I_{LD} \rightarrow I_{FD}$ and ψ_{F2L} learns mapping function from $I_{FD} \rightarrow I_{LD}$. The network ψ_{L2F} takes I_{LD} as input and predicts $\hat{I_{FD}}$ so as to fool D_F into classifying it as real while D_F attempts to discriminate between predicted $\hat{I_{FD}}$ and real I_{FD}. Similarly, the network ψ_{F2L} takes I_{FD} as an input and predicts $\hat{I_{LD}}$ so as to fool D_L into classifying it as real while D_L attempts to discriminate between predicted $\hat{I_{LD}}$ and real I_{LD}. The networks ψ_{L2F} and ψ_{F2L} are two generators following the architecture of the PMVA-Unet model as discussed in section 5.3. The networks D_L and D_F are patch discriminators as proposed in Pix2Pix [45]. Cross entropy loss between real and fake of respective domains is used to train the discriminator networks. Cyclic loss enforces $\psi_{F2L}(\psi_{L2F}(I_{LD})) \approx I_{LD}$ and $\psi_{L2F}(\psi_{F2L}(I_{FD})) \approx I_{FD}$. The overall objective function (L_{un-sup}) for ψ_{L2F} combines the following losses (explained in detail in section

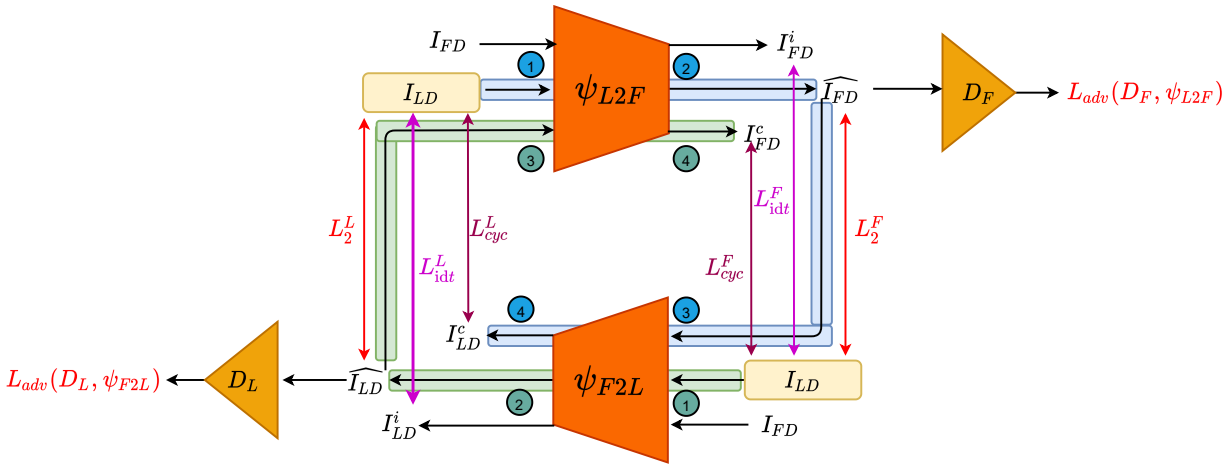


Figure 5.6: ψ_{L2F} and ψ_{F2L} are the generators to transform LDCT to FDCT and vice-versa. D_L and D_F denote the Discriminators. Blue and Green shadow lines show the direction in which input moves for calculating both the terms of cyclic loss: $L_{cyc}(\psi_{L2F}, \psi_{F2L}) = L^L_{cyc} + L^F_{cyc}$. Similarly, Identity loss is sum of two terms represented in purple color: $L_{idt}(\psi_{L2F}, \psi_{F2L}) = L^L_{idt} + L^F_{idt}$. L_2 loss is computed only in case of semi-supervised training when paired data is passed through the network. Each of the generators are trained with respective L_2 loss represented by red arrow marks. $L_{adv}(D_F, \psi_{L2F})$ and $L_{adv}(D_L, \psi_{F2L})$ are used to train ψ_{L2F} and ψ_{F2L} respectively.

2.5) to learn the mapping between un-paired LDCT and FDCT data as defined below:

$$L_{un-sup}(\psi_{L2F}) = L_{adv}(D_F, \psi_{L2F}) + \lambda_1 L_{cyc}(\psi_{L2F}, \psi_{F2L}) + \lambda_2 L_{idt}(\psi_{L2F}, \psi_{F2L}) \quad (5.3.2)$$

5.3.3 Semi-Supervised training on semi-paired data:

In case of semi-supervised training scenario, we have used some set of paired along with unpaired LDCT and FDCT data. Out of the ten subjects available in dataset-I, six scans for training, two corresponding LDCT and FDCT Scans are taken from the same patient (paired data). An additional 4 FDCT Scans are chosen from 4 different patients, making a total of 2 LDCT and 6 FDCT Scans for training. Here, 30% paired data has been utilized. The network learns the mapping function from paired and unpaired data is expected to boost the performance by utilizing identity loss and augmented with cyclic snd adversarial losses. Similar to the previous section, CycleGAN has been used for training. However,

we propose a technique of flowing different losses through the network depending upon different types of data (paired/ unpaired) that is being passed through the network. Loss for training ψ_{L2F} is shown below:

$$L_{semi-sup}^{un-paired}(\psi_{L2F}) = L_{adv}(D_F, \psi_{L2F}) + \lambda_1 L_{cyc}(\psi_{L2F}, \psi_{F2L}) + \lambda_2 L_{idt}(\psi_{L2F}, \psi_{F2L}) \quad (5.3.3)$$

$$L_{semi-sup}^{paired}(\psi_{L2F}) = L_{adv}(D_F, \psi_{L2F}) + L_2(\psi_{L2F}) \quad (5.3.4)$$

In each epoch, we train the network both on paired data with $L_{semi-sup}^{paired}$ and on un-paired data with $L_{semi-sup}^{un-paired}$. L_2 loss is only utilized for paired data setting because in unpaired setting there are no corresponding pairs for one-to-one comparison.

5.4 Experimental Analysis

Dataset I and Dataset II both are available in Dicom format. We have used the Pydicom library for data preprocessing and to convert the image into a Numpy array. The images are normalized between 0 and 1 before passing it through the network. The reconstruction quality of CT Scan images is evaluated quantitatively using three evaluation metrics: (i) Peak Signal to Noise Ratio (ii) MSE - Mean Square Error and (iii) SSIM - Structural Similarity Index. All the above metrics are described in Section 3.3. 3. RMSprop is used to optimize the network parameters. The networks are implemented using Tensorflow 2.0, and are trained/validated over a workstation using NVIDIA TITAN V GPU Card.

5.4.1 Comparative Analysis on Dataset-I

Choosing exactly the same ten patients scans published by the Grand Challenge in 2016 from the 50 scans that are curated in the library now is not possible. This is because the identity of them is not revealed. Hence, in order to have a fair comparison, we randomly

Model	PSNR	MSE	SSIM
SACNN [9]	31.90	0.02538	0.93913
Unet	36.80	0.01396	0.97769
PMVA-Unet (Proposed)	**37.20**	**0.01393**	**0.98652**

Table 5.2: Proposed architecture performs better than the SOTA and Unet comparatively on Dataset-I with respect to PSNR, MSE and SSIM metrics.

split the 50 scans into five sets of 10 scans each. Then we have ran five PMVA-Unet models (proposed method) - M1, M2, M3, M4, M5 on each set, respectively and chose that set over which PMVA-Unet has obtained the least performance metric score. This is good to ensure that our least is comparable to SOTA. For the same reason, we implemented the SOTA - SACNN(CNN-SA-AE version) [9] and ran it over the chosen set of data. Our code along with trained model file has been publicly available in Github [5]. Of the ten scans, we used six scans for training (898 slices), two scans for validation (233 slices) and two scans for testing (299 slices). This experiment is done in a Supervised setup where each of the six scans of LDCT and six scans of FDCT has been chosen from the same patient.

Comparison with existing work is shown in Table 5.2, and one can see that the PSNR of the proposed method performs significantly better (by 5dB) than SOTA. The rise in metrics can be attributed to showing the whole image to the network instead of small pieces and efficiently computing attention. As a part of the ablation study, we also show results with Unet [5] where all the feature extraction layers are implemented using simple conv layers. Code and trained models are available on Github [6]. Although the increase in PSNR is less, qualitatively reconstructed images in Figure 5.7 shows a lot of improvement.

As seen in the blue circles of Figure 5.7(c), the predicted image from the SACNN method has border artefacts. SACNN is trained to take an input of size $64 * 64 * 3$ and

[5] https://github.com/s3pi/Enhancing-Lowdose-CT-Scan/tree/main/Supervised/SACNN
[6] https://github.com/s3pi/Enhancing-Lowdose-CT-Scan/tree/main/Supervised/Unet

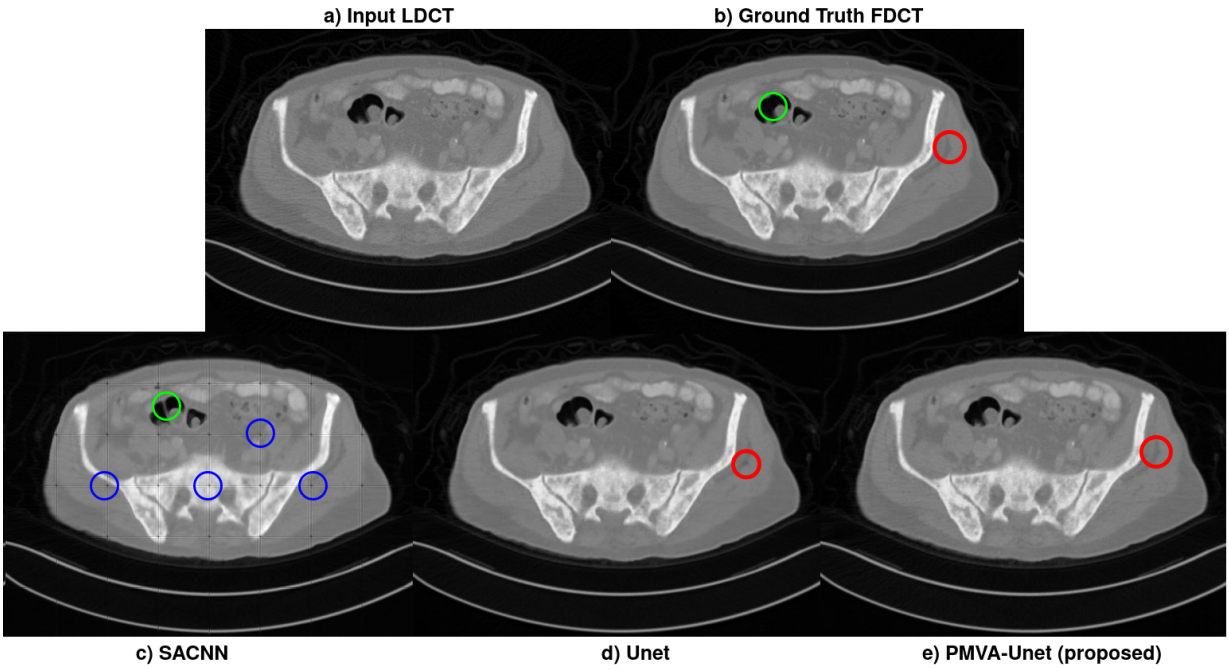


Figure 5.7: Qualitative analysis of proposed methodology compared to SOTA and Unet. This is performed on case L058 from NIH-AAPM-Mayo Clinic dataset available in TCIA. (a) and (b) show the input and ground truth respectively. (c), (d) and (e) are the predicted outputs from SACNN(SOTA), Unet(for ablation study) and proposed method respectively. The Red circle on (d)Unet shows an artefact while the (e)proposed network does not generate that artefact. Blue circles in (c)SACNN show border artefacts. Green circle show the artefact created by SACNN while absent in the ground truth.

output an image of size $64 * 64 * 1$ where the dimensions indicate height, width and depth, respectively. Since no FC layers are used in SACNN, we can test it on the entire image of size $512 * 512 * 3$. However, we are able to test it only on a maximum patch size of $64 * 64 * 3$ and augment the non-overlapping patches because of memory constraints even with the 11GB RAM of NVIDIA TITAN V. The red circle in Figure 5.7(d) indicates an artefact generated by Unet which does not appear in the image reconstructed by the proposed method Figure 5.7(e).

Paradigm	Model	PSNR
Supervised	PMVA-Unet	37.20
Semi Supervised	PMVA-Unet as generator with different losses Paired : Adversarial Loss + L2 Loss Unpaired : Cyclic Loss + Adversarial Loss + Identity Loss.	31.16
Un-Supervised	Cycle GAN with PMVA-Unet as generator	29.84

Table 5.3: Supervised training performs better than semi-supervised and un-supervised training in the same order in Dataset-I.

5.4.2 Analysis on Dataset-I under various settings

Results for all the three paradigms of training on Dataset I is summarized in Table 5.3. For comparison purposes, in all three cases, validation and test scans are kept the same. As expected, PSNR obtained in a supervised scenario surpasses the results obtained by Semi-Supervised and Unsupervised in the same order. It is easier for the network to approximate a mapping function when more paired images are used for training. As shown in red and blue circles of Figure 5.8(d), unsupervised training could not reconstruct many details. However, it got the overall structure right. The circles also show improved and detailed reconstruction in the supervised over semi-supervised training paradigm.

5.4.3 Comparative Analysis on Dataset-II

Model	PSNR	MSE	SSIM
SACNN[9]	22.45	0.044980	0.90523
Unet	27.02	0.044520	0.92577
PMVA-Unet (Proposed)	**27.35**	**0.044518**	**0.93432**

Table 5.4: Proposed architecture performs better than the SOTA and Unet comparatively on Dataset-II with respect to PSNR, MSE and SSIM metrics.

Dataset - II is a clinical data obtained from PGIMER Chandigarh. Data is acquired from a Siemens scanner with 70kV radiation for low-dose and 100kV for full-dose im-

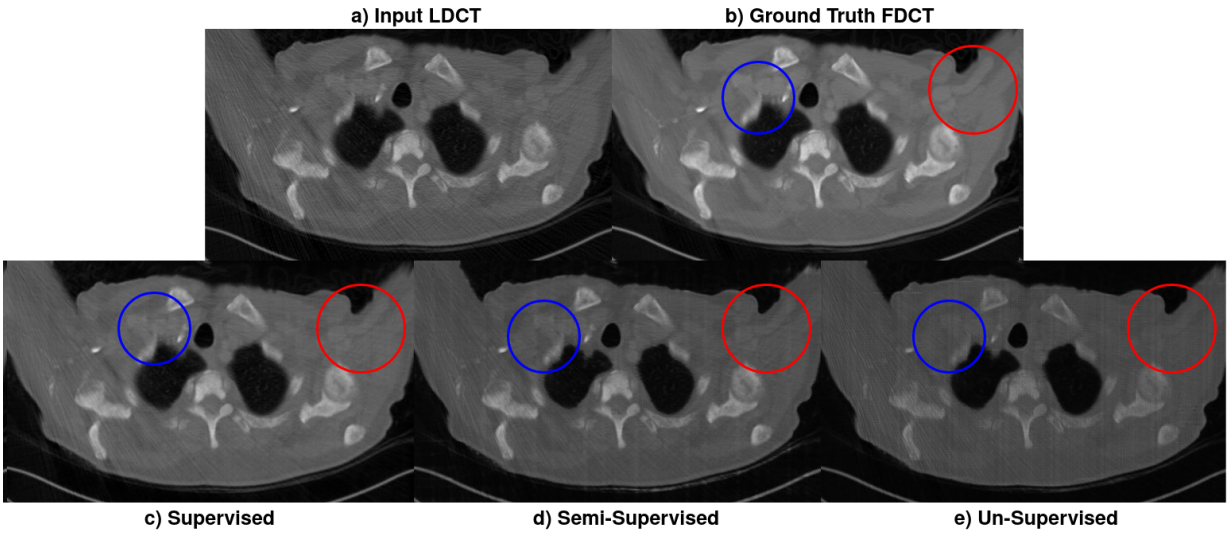


Figure 5.8: Qualitative analysis of proposed methodology on L058 subject from NIH-AAPM-Mayo clinic dataset available in TCIA, is compared among (c) supervised training paradigm with paired data, (d) semi-supervised training paradigm with semi-paired data and (e) un-supervised training paradigm with un-paired data. Red and Blue circles show an improved reconstruction in the specified regions, in case of supervised over semi-supervised over un-supervised.

ages. We then implemented the SOTA - SACNN(CNN-SA-AE version) [9] and Unet[5] and trained over this data. Code along with trained model file is available in the same Github location as above. Analysis on Dataset-II has been done similar to Dataset-I. The comparitive results of comparison with existing work can be seen in Table 5.4. We can see that the PSNR of the proposed method performs better than SOTA. Although the improvement is not significant the qualitative improvement in structural details achieved by the proposed method is very stark. There have been works in the past which showed significant improvement in quality but not in PSNR as in Red-CNN [95] had better PSNR and SSIM than SACNN while the later achieved better quality reconstruction. Similar examples also exist in case of natural images, for super resolution task by SRGAN([111]) and compression task by [112], and they argue that standard quantitative measures such as PSNR and SSIM fail to capture and accurately assess image quality with respect to the human visual system. Since PSNR is an overall metric, structural improvements to an

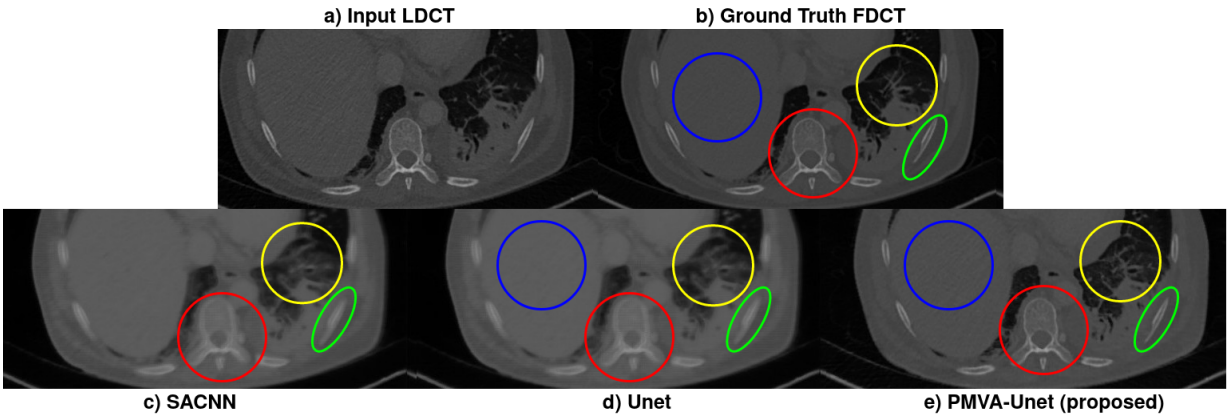


Figure 5.9: Qualitative analysis of proposed methodology compared to SOTA and Unet is discussed on Test-Case-1 from PGI-Chandigarh clinical dataset. (a) and (b) show the input and ground truth, respectively. (c), (d) and (e) are the predicted outputs from SACNN(SOTA), Unet(for ablation study) and proposed method, respectively. Red, Green and Yellow circled regions show that the proposed method reconstructed the finer details much closer to the ground truth than SACNN. the Blue circle shows that the proposed method failed to smoothen the region, as well as Unet, did.

image locally may not contribute significantly to the overall PSNR. However, we observe a decrease in PSNR value compared to the one obtained on Dataset-I. This is majorly due to the high resolution images in Dataset-II with pixel intensities ranging from 0 to 4000. Since the dynamic range got extended its reconstruction becomes more challenging.

The yellow circle on the image from the proposed method in Figure 5.9(e) shows that the reconstruction of fine level details are much more accurate and visually better than SACNN and Unet, which smudged those details. Red and Yellow circled regions show that edges of the bones are much clearer by the proposed method than Unet and SACNN. We have observed that Unet over smoothened all the three colour regions mentioned above. While the blue circle shows that the proposed method failed to smoothen the liver region, Unet did better. Visually one can observe that the overall quality of the reconstructed image is much better by the proposed method on Dataset-II than the other two methods.

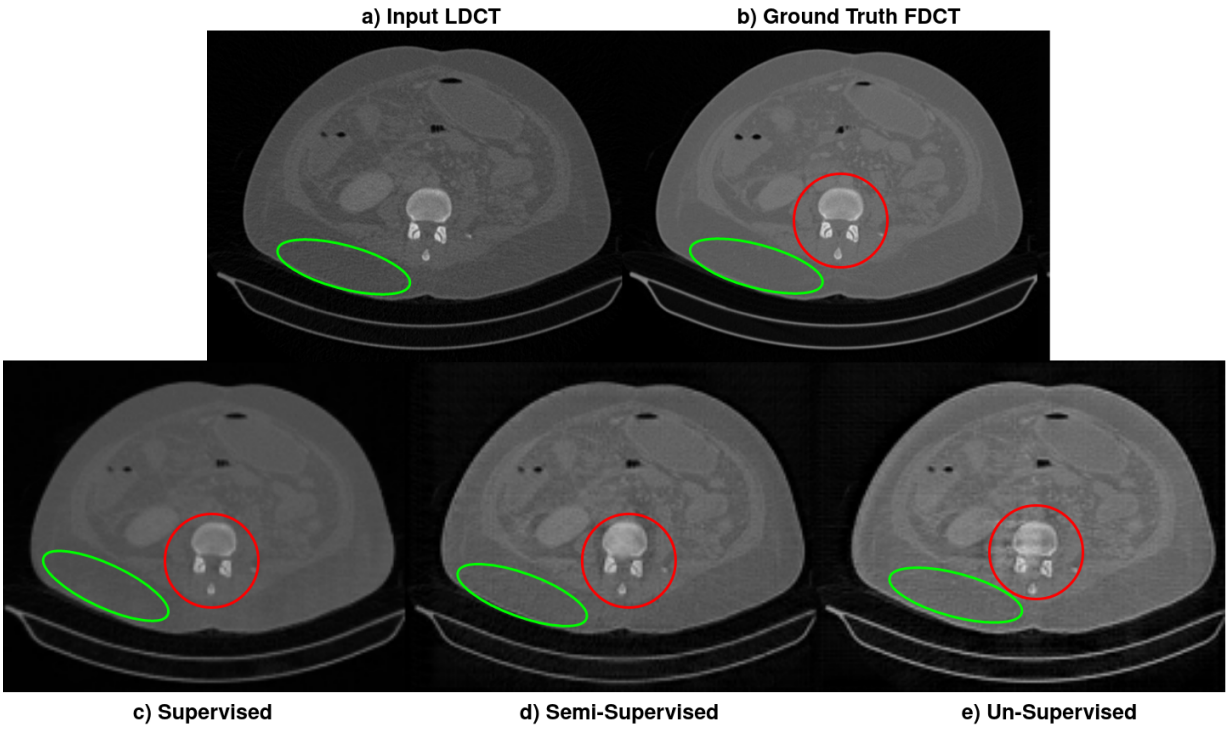


Figure 5.10: Qualitative analysis of proposed methodology is discussed on Test-Case-2 from PGI-Chandigarh clinical dataset. Comparison is shown among (c) supervised training paradigm with paired data, (d) semi-supervised training paradigm with semi-paired data and (e) un-supervised training paradigm with un-paired data. Red and Green circles show an improved reconstruction in the specified regions, in case of supervised over semi-supervised over un-supervised.

5.4.4 Analysis on Dataset-II under various settings

As shown in Table 5.5, we can observe decreasing PSNR values in Supervised, Semi-supervised and Unsupervised cases in the same order similar to behaviour in Dataset-I. The red circle in Figure 5.10(e) shows that reconstruction in the case of unpaired data is noisier than semi-paired and paired. The green circle shows that the supervised setup smoothened the region better than the other setups.

Paradigm	Model	PSNR
Supervised	PMVA-Unet	29.0
Semi Supervised	PMVA-Unet as generator with different losses Paired : Adversarial Loss + L2 Loss Unpaired : Cyclic Loss + Adversarial Loss + Identity Loss.	24.9
Un-Supervised	Cycle GAN with PMVA-Unet as generator	24.14

Table 5.5: Supervised training performs better than semi-supervised and un-supervised training in the same order because the number of paired images seen by the network decreases with each case in Dataset-II.

5.5 Conclusion

This chapter introduces a novel method of denoising low-dose CT Scans. Contributions of this chapter are as follows - 1) The proposed technique can take the entire CT Scan image as an input instead of small pieces of it. At the same time, the technique is also computationally efficient with patch-wise attention. 2) Transformer network is adapted to work on CT image features. 3) We show analysis in case of un-supervised training paradigm and provide a technique of flowing different losses through the same network based on the data setting in the semi-supervised training paradigm. 4) We show results on a dataset available in a public library and clinical data from PGI-Chandigarh hospital. Results are compared with SOTA and ablated models to find an increase in performance quantitatively. Furthermore, the proposed method achieves significant improvement in perceptive quality with images reconstructed sharply.

5.6 Acknowledgements

We thank Dr Pankaj Gupta (Associate Professor) [7] for providing us with Abdomen CT Scan data.

[7] https://www.researchgate.net/profile/Pankaj-Gupta-10

Chapter 6

Summary and Future Work

6.1 Summary

This chapter aims to summarize the contributions made in synthesising MRI, automatically generating X-ray report and denoising low dose CT Scan that can help medical image analysis to reap the benefits like reduction in acquisition time, increase the comfort of patients and doctors by automation and decrease the radiation hazards. We explored novel deep learning architectures and methods for image analysis which can help towards addressing these problems. The proposed methods can be summarized as follows:

- A deep convolutional, Encoder-Decoder based deep learning architecture is built to reconstruct T2 modality of MRI image from T1 modality of MRI image. This synthesis technique helps to reduce the acquisition time and thereby alleviate comfort and decrease the per-person cost.
- We proposed an attention-based deep neural network to generate a report for X-rays automatically. Automating X-ray report generation can be useful for quick and automated mass screening.

- CT Scanners induce X-ray radiation to capture images of internal organs. A higher radiation dosage leads to clearer images than images generated with a lower dosage of radiation. Nevertheless, higher doses of radiation have harmful effects. Hence, we proposed an architecture that computes visual attention across non-overlapping patches to denoise the low dose CT scans. Furthermore, we proposed techniques to address the availability of only fewer or no paired data to learn the image transformation function.

6.2 Future Work

This thesis improves over the state-of-the-art systems on the chosen tasks. However, there are many areas with scope for improvement, and a few of them are enumerated below:

- In this thesis, we worked on synthesising one image modality to another. As part of future work, we would like to explore one-to-many and many-to-many transformations.

- Automating X-ray report generation involved producing text reports from an image alone. We would further like to explore using image data along with clinical observations to generate a report.

- Many medical images like CT and MRI Scans are very large, and have a very high range of pixel intensities varying from 0 to 4000 unit values. This may restrict the batch size and model size due to memory constraints. Thus, efficiency considerations for larger volumes within the memory limitations can be part of future work.

List of Publications

3. Preethi Srinivasan, Daksh Thapar, Aditya Nigam and Arnav Bhavsar **"Hierarchical X-Ray Report Generation via Pathology tags and Multi Head Attention."** Accepted in 15th Asian Conference on Computer Vision (ACCV), Japan, 2020.

2. Preethi Srinivasan, Prabhjot Kaur, Aditya Nigam and Arnav Bhavsar **"Semantic features aided multi scale reconstruction of inter-modality MR images."** Accepted in 33rd IEEE CBMS International Symposium on Computer based medical systems, USA, 2020.

1. Preethi Srinivasan, Prabhjot Kaur, Aditya Nigam and Arnav Bhavsar **"An MRI Inter-Modality Reconstruction Network using Multiscale Feature Transformation"** Accepted in WiML Workshop, NIPS 2019, Vancouver, Canada, 2019.

Bibliography

[1] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017.

[2] "Understanding and implementing cyclegan in tensorflow," https://hardikbansal.github.io/CycleGANBlog/.

[3] L. Xiang, Y. Chen, W. Chang, Y. Zhan, W. Lin, Q. Wang, and D. Shen, "Ultra-fast T2-weighted MR reconstruction using complementary T1-weighted information," in *International Conference on Medical image computing and computer-assisted intervention*, 2018.

[4] R. Bitar, G. Leung, R. Perng, S. Tadros, A. R. Moody, J. Sarrazin, C. McGregor, M. Christakis, S. Symons, A. Nelson *et al.*, "Mr pulse sequences: what every radiologist wants to know but is afraid to ask," *Radiographics*, vol. 26, no. 2, pp. 513–537, 2006.

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

[6] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou,

"Transunet: Transformers make strong encoders for medical image segmentation," *arXiv preprint arXiv:2102.04306*, 2021.

[7] S. Bhadra, W. Zhou, and M. A. Anastasio, "Medical image reconstruction with image-adaptive priors learned by use of generative adversarial networks," in *Medical Imaging 2020: Physics of Medical Imaging*, 2020.

[8] X. Deng and P. L. Dragotti, "Deep convolutional neural network for multi-modal image restoration and fusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[9] M. Li, W. Hsu, X. Xie, J. Cong, and W. Gao, "Sacnn: Self-attention convolutional neural network for low-dose ct denoising with self-supervised perceptual loss network," *IEEE Transactions on Medical Imaging*, vol. 39, no. 7, pp. 2289–2301, 2020.

[10] K. Thurnhofer-Hemsi, E. López-Rubio, E. Domínguez, R. M. Luque-Baena, and N. Roé-Vellvé, "Deep learning-based super-resolution of 3d magnetic resonance images by regularly spaced shifting," *Neurocomputing*, vol. 398, pp. 314–327, 2020.

[11] J. Ma, J. Yu, S. Liu, L. Chen, X. Li, J. Feng, Z. Chen, S. Zeng, X. Liu, and S. Cheng, "Pathsrgan: Multi-supervised super-resolution for cytopathological images using generative adversarial network," *IEEE Transactions on Medical Imaging*, vol. 39, no. 9, pp. 2920–2930, 2020.

[12] M. Yurt, S. U. Dar, A. Erdem, E. Erdem, K. K. Oguz, and T. Çukur, "mustgan: multi-stream generative adversarial networks for mr image synthesis," *Medical Image Analysis*, vol. 70, p. 101944, 2021.

[13] C. Wang, G. Yang, G. Papanastasiou, S. A. Tsaftaris, D. E. Newby, C. Gray, G. Macnaught, and T. J. MacGillivray, "Dicyc: Gan-based deformation invariant

cross-domain information fusion for medical image synthesis," *Information Fusion*, vol. 67, pp. 147–160, 2021.

[14] B. Cao, H. Zhang, N. Wang, X. Gao, and D. Shen, "Auto-gan: self-supervised collaborative learning for medical image synthesis," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[15] Y. Fu, Y. Lei, T. Wang, W. J. Curran, T. Liu, and X. Yang, "Deep learning in medical image registration: a review," *Physics in Medicine & Biology*, vol. 65, no. 20, p. 20TR01, 2020.

[16] H. R. Boveiri, R. Khayami, R. Javidan, and A. Mehdizadeh, "Medical image registration using deep neural networks: A comprehensive review," *Computers Electrical Engineering*, vol. 87, p. 106767, 2020.

[17] J. Zhu, Y. Li, Y. Hu, K. Ma, S. K. Zhou, and Y. Zheng, "Rubik's cube+: A self-supervised feature learning framework for 3d medical image analysis," *Medical Image Analysis*, vol. 64, p. 101746, 2020.

[18] S. Budd, E. C. Robinson, and B. Kainz, "A survey on active learning and human-in-the-loop deep learning for medical image analysis," *Medical Image Analysis*, vol. 71, 2021.

[19] T. Mitchell, *Machine Learning*, 1st ed. McGraw Hill, 1997.

[20] W. S. McCulloch and W. H. Pitts, "A logical calculus of the ideas immanent in nervous activity," in *The bulletin of mathematical biophysics*, vol. 5, 1943, pp. 115–133.

[21] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, pp. 386–408, 1958.

[22] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[23] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning, ICML, Lille, France*, 2015, pp. 448–456.

[24] Y. Li, Y. Song, and J. Luo, "Improving pairwise ranking for multi-label image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[25] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, 1st ed. MIT Press, 2016.

[26] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017.

[28] "Transformer architecture positional encoding," https://kazemnejad.com/blog/.

[29] O. Commowick, F. Cervenansky, and R. Ameli, "Msseg challenge proceedings: multiple sclerosis lesions segmentation challenge using a data management and processing infrastructure," in *International Conference on Medical image computing and computer-assisted intervention*, 2016.

[30] "https://www.humanconnectome.org/study/hcp-young-adult/document/1200-subjects-data-release."

[31] R. A. Poldrack, J. A. Mumford, and T. E. Nichols, *Handbook of Functional MRI Data Analysis*. Cambridge University Press, 2011.

[32] R. Vemulapalli, H. V. Nguyen, and S. K. Zhou, "Unsupervised cross-modal synthesis of subject-specific scans," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[33] C. Alkan, J. Cocjin, and A. Weitz, "Magnetic resonance contrast prediction using deep learning," *Google Scholar*, 2016.

[34] J. Schlemper, J. Caballero, J. V. Hajnal, A. N. Price, and D. Rueckert, "A deep cascade of convolutional neural networks for dynamic MR image reconstruction," *IEEE Transactions on Medical Imaging*, vol. 37, no. 2, pp. 491–503, 2018.

[35] Y. Huang, L. Shao, and A. F. Frangi, "Cross-modality image synthesis via weakly coupled and geometry co-regularized joint dictionary learning," *IEEE Transactions on Medical Imaging*, vol. 37, no. 3, pp. 815–827, 2017.

[36] A. Chartsias, T. Joyce, M. V. Giuffrida, and S. A. Tsaftaris, "Multimodal mr synthesis via modality-invariant latent representation," *IEEE Transactions on Medical Imaging*, vol. 37, no. 3, pp. 803–814, 2018.

[37] A. Sharma and G. Hamarneh, "Missing mri pulse sequence synthesis using multi-modal generative adversarial network," 2019.

[38] S. U. Dar, M. Yurt, L. Karacan, A. Erdem, E. Erdem, and T. Çukur, "Image synthesis in multi-contrast mri with conditional generative adversarial networks," *IEEE Transactions on Medical Imaging*, vol. 38, no. 10, pp. 2375–2388, 2019.

[39] D. Kroon and C. H. Slump, "MRI modalitiy transformation in demon registration,"

in *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2009, pp. 963–966.

[40] H. V. Nguyen, S. K. Zhou, and R. Vemulapalli, "Cross-domain synthesis of medical images using efficient location-sensitive deep network," in *International Conference on Medical image computing and computer-assisted intervention*, 2015.

[41] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," *CoRR*, vol. abs/1603.06937, 2016.

[42] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[43] S. M. Smith, "Fast robust automated brain extraction," *Human Brain Mapping*, vol. 17, no. 3, pp. 143–155, 2002.

[44] M. Jenkinson and S. Smith, "A global optimisation method for robust affine registration of brain images," *Medical Image Analysis*, vol. 5, no. 2, pp. 143 – 156, 2001.

[45] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[46] L. Delrue, R. Gosselin, B. Ilsen, A. Van Landeghem, J. de Mey, and P. Duyck, "Difficulties in the interpretation of chest radiography," in *Comparative interpretation of CT and standard radiography of the chest*. Springer, 2011.

[47] B. Jing, P. Xie, and E. Xing, "On the automatic generation of medical imaging reports," *ACL*, 2018.

[48] X. Wang, Y. Peng, L. Lu, Z. Lu, and R. M. Summers, "Tienet: Text-image embedding network for common thorax disease classification and reporting in chest x-rays," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[49] J. Johnson, A. Karpathy, and L. Fei-Fei, "Densecap: Fully convolutional localization networks for dense captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[50] D. Demner-Fushman, M. D. Kohli, M. B. Rosenman, S. E. Shooshan, L. Rodriguez, S. Antani, G. R. Thoma, and C. J. McDonald, "Preparing a collection of radiology examinations for distribution and retrieval," *Journal of the American Medical Informatics Association*, vol. 23, no. 2, pp. 304–310, 2016.

[51] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. Summers, "Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[52] L. Yao, E. Poblenz, D. Dagunts, B. Covington, D. Bernard, and K. Lyman, "Learning to diagnose from scratch by exploiting dependencies among labels," *arXiv preprint arXiv:1710.10501*, 2017.

[53] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya *et al.*, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *arXiv preprint arXiv:1711.05225*, 2017.

[54] P. Kisilev, E. Walach, E. Barkan, B. Ophir, S. Alpert, and S. Y. Hashoul, "From medical image to automatic medical report generation," *IBM Journal of Research and Development*, vol. 59, no. 2/3, pp. 2–1, 2015.

[55] H.-C. Shin, K. Roberts, L. Lu, D. Demner-Fushman, J. Yao, and R. M. Summers, "Learning to read chest x-rays: Recurrent neural cascade model for automated image annotation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[56] Z. Zhang, Y. Xie, F. Xing, M. McGough, and L. Yang, "Mdnet: A semantically and visually interpretable medical image diagnosis network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[57] Y. Li, X. Liang, Z. Hu, and E. P. Xing, "Hybrid retrieval-generation reinforced agent for medical image report generation," in *Advances in neural information processing systems*, 2018.

[58] Y. Xiong, B. Du, and P. Yan, "Reinforced transformer for medical image captioning," in *International Workshop on Machine Learning in Medical Imaging*, 2019.

[59] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[60] J. Weston, S. Bengio, and N. Usunier, "Wsabie: Scaling up to large vocabulary image annotation," in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[61] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1338–1351, 2006.

[62] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[63] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*, 2016.

[64] D. Thapar, G. Jaswal, A. Nigam, and C. Arora, "Gait metric learning siamese network exploiting dual of spatio-temporal 3d-cnn intra and lstm based inter gait-cycle-segment features," *Pattern Recognition Letters*, vol. 125, pp. 646–653, 2019.

[65] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002.

[66] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[67] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, 2015.

[68] G. Liu, T.-M. H. Hsu, M. McDermott, W. Boag, W.-H. Weng, P. Szolovits, and M. Ghassemi, "Clinically accurate chest x-ray report generation," *arXiv preprint arXiv:1904.02633*, 2019.

[69] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[70] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: Adaptive

attention via a visual sentinel for image captioning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[71] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, "Self-critical sequence training for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[72] L. Fu, T.-C. Lee, S. M. Kim, A. M. Alessio, P. E. Kinahan, Z. Chang, K. Sauer, M. K. Kalra, and B. De Man, "Comparison between pre-log and post-log statistical models in ultra-low-dose ct reconstruction," *IEEE transactions on medical imaging*, vol. 36, no. 3, pp. 707–720, 2016.

[73] R. Smith-Bindman, J. Lipson, R. Marcus, K.-P. Kim, M. Mahesh, R. Gould, A. B. De González, and D. L. Miglioretti, "Radiation dose associated with common computed tomography examinations and the associated lifetime attributable risk of cancer," *Archives of internal medicine*, vol. 169, no. 22, pp. 2078–2086, 2009.

[74] A. B. De González, M. Mahesh, K.-P. Kim, M. Bhargavan, R. Lewis, F. Mettler, and C. Land, "Projected cancer risks from computed tomographic scans performed in the united states in 2007," *Archives of internal medicine*, vol. 169, no. 22, pp. 2071–2077, 2009.

[75] D. M. et al., "Projection space denoising with bilateral filtering and ct noise modeling for dose reduction in ct," *Medical Physics*, vol. 36, no. 11, pp. 4911–4919, 2009.

[76] J. H. M. Balda and B. Heismann, "Ray contribution masks for structure adaptive sinogram filtering," *IEEE Transactions on Medical Imaging*, vol. 31, no. 6, pp. 1228–1239, 2012.

[77] H. L. J. Wang, T. Li and Z. Liang, "Penalized weighted least-squares approach to sinogram noise reduction and image reconstruction for low-dose x-ray computed

tomography," *IEEE Transactions on Medical Imaging*, vol. 25, no. 10, pp. 1272–1283, 2006.

[78] E. Y. Sidky and X. Pan, "Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization," *Physics in Medicine and Biology*, vol. 53, no. 17, pp. 4777–4807, 2008.

[79] Z. Tian, X. Jia, K. Yuan, T. Pan, and S. B. Jiang, "Low-dose CT reconstruction via edge-preserving total variation regularization," *Physics in Medicine and Biology*, vol. 56, no. 18, pp. 5949–5967, 2011.

[80] Y. Liu, J. Ma, Y. Fan, and Z. Liang, "Adaptive-weighted total variation minimization for sparse data toward low-dose x-ray computed tomography image reconstruction," *Physics in Medicine and Biology*, vol. 57, no. 23, pp. 7923–7956, nov 2012.

[81] Y. Zhang, W. Zhang, Y. Lei, and J. Zhou, "Few-view image reconstruction with fractional-order total variation," *J. Opt. Soc. Am. A*, vol. 31, no. 5, pp. 981–995, 2014.

[82] Y. Chen, D. Gao, C. Nie, L. Luo, W. Chen, X. Yin, and Y. Lin, "Bayesian statistical reconstruction for low-dose x-ray computed tomography using an adaptive-weighting nonlocal prior," *Computerized Medical Imaging and Graphics*, vol. 33, no. 7, pp. 495–500, 2009.

[83] J. Ma, H. Zhang, Y. Gao, J. Huang, Z. Liang, Q. Feng, and W. Chen, "Iterative image reconstruction for cerebral perfusion CT using a pre-contrast scan induced edge-preserving prior," *Physics in Medicine and Biology*, vol. 57, no. 22, pp. 7519–7542, oct 2012.

[84] Y. Zhang, Y. Xi, Q. Yang, W. Cong, J. Zhou, and G. Wang, "Spectral ct reconstruc-

tion with image sparsity and spectral mean," *IEEE Transactions on Computational Imaging*, vol. 2, no. 4, pp. 510–523, 2016.

[85] Q. Xu, H. Yu, X. Mou, L. Zhang, J. Hsieh, and G. Wang, "Low-dose x-ray ct reconstruction via dictionary learning," *IEEE Transactions on Medical Imaging*, vol. 31, no. 9, pp. 1682–1697, 2012.

[86] Y. Zhang, X. Mou, G. Wang, and H. Yu, "Tensor-based dictionary learning for spectral ct reconstruction," *IEEE Transactions on Medical Imaging*, vol. 36, no. 1, pp. 142–154, 2017.

[87] J. M. et al., "Low-dose computed tomography image restoration using previous normal-dose scan," *Medical Physics*, vol. 38, no. 10, pp. 5713–5731, 2011.

[88] Z. Li, L. Yu, J. D. Trzasko, D. S. Lake, D. J. Blezek, J. G. Fletcher, C. H. McCollough, and A. Manduca, "Adaptive nonlocal means filtering based on local noise level for ct denoising," *Medical physics*, vol. 41, no. 1, p. 011908, 2014.

[89] B. B. Z. S. Kelm, D. Blezek and B. J. Erickson, "Optimizing non-local means for denoising low dose ct," *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 662–665, 2009.

[90] M. E. M. Aharon and A. Bruckstein, "K-svd: An algorithm for designing over complete dictionaries for sparse representation," *IEEE Transactions on Signal Processing.*, vol. 54, no. 11, pp. 4311–4322, 2006.

[91] Y. C. et al., "Improving abdomen tumor low-dose ct images using a fast dictionary learning based processing," *Physics in Medicine and Biology*, vol. 58, no. 16, pp. 5803–5820, 2013.

[92] J. G. A. S. P. F. Feruglio, C. Vinegoni and R. Weissleder, "Block matching 3d random noise fil tering for absorption optical projection tomography," *Physics in Medicine and Biology*, vol. 55, no. 18, pp. 5401–5415, 2010.

[93] J. W. K. Sheng, S. Gou and S. X. Qi, "Denoised and texture enhanced mvct to improve soft tissue conspicuity," *Medical Physics*, vol. 41, no. 10, 2014.

[94] D. Kang, P. Slomka, R. Nakazato, J. Woo, D. S. Berman, C.-C. J. Kuo, and D. Dey, "Image denoising of low-radiation dose coronary ct angiography by an adaptive block-matching 3d algorithm," in *Medical Imaging 2013: Image Processing*, 2013.

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

[96] D. Wu, K. Kim, G. E. Fakhri, and Q. Li, "A cascaded convolutional neural network for x-ray low-dose ct image denoising," 2017.

[97] E. Kang, J. Min, and J. C. Ye, "A deep convolutional neural network using directional wavelets for low-dose x-ray ct reconstruction," *Medical Physics*, vol. 44, no. 10, p. e360–e375, 2017.

[98] H. Chen, Y. Zhang, W. Zhang, P. Liao, K. Li, J. Zhou, and G. Wang, "Low-dose ct denoising with convolutional neural network," in *2017 IEEE 14th International Symposium on Biomedical Imaging*, 2017.

[99] Q. Yang, P. Yan, M. K. Kalra, and G. Wang, "Ct image denoising with perceptive deep neural networks," *arXiv preprint arXiv:1702.07019*, 2017.

[100] M. Gholizadeh-Ansari, J. Alirezaie, and P. Babyn, "Deep learning for low-dose ct

denoising using perceptual loss and edge detection layer," *Journal of Digital Imaging*, vol. 33, 09 2019.

[101] C. You, Q. Yang, H. Shan, L. Gjesteby, G. Li, S. Ju, Z. Zhang, Z. Zhao, Y. Zhang, W. Cong, and G. Wang, "Structurally-sensitive multi-scale deep neural network for low-dose ct denoising," *IEEE Access*, vol. 6, pp. 41 839–41 855, 2018.

[102] H. Shan, Y. Zhang, Q. Yang, U. Kruger, M. K. Kalra, L. Sun, W. Cong, and G. Wang, "3-d convolutional encoder-decoder network for low-dose ct via transfer learning from a 2-d trained network," *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, p. 1522–1534, Jun 2018.

[103] L. Wei, Y. Lin, and W. Hsu, "Using a generative adversarial network for ct normalization and its impact on radiomic features," 2020.

[104] J. M. Wolterink, T. Leiner, M. A. Viergever, and I. Išgum, "Generative adversarial networks for noise reduction in low-dose ct," *IEEE Transactions on Medical Imaging*, vol. 36, no. 12, pp. 2536–2545, 2017.

[105] X. Yi and P. Babyn, "Sharpness-aware low-dose ct denoising using conditional generative adversarial network," *Journal of Digital Imaging*, vol. 31, no. 5, p. 655–669, 2018.

[106] Q. Yang, P. Yan, Y. Zhang, H. Yu, Y. Shi, X. Mou, M. K. Kalra, Y. Zhang, L. Sun, and G. Wang, "Low-dose ct image denoising using a generative adversarial network with wasserstein distance and perceptual loss," *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, pp. 1348–1357, 2018.

[107] M. Li, S. Shen, W. Gao, W. Hsu, and J. Cong, "Computed tomography image enhancement using 3d convolutional neural network," 2018.

[108] C. You, W. Cong, M. W. Vannier, P. K. Saha, E. A. Hoffman, G. Wang, G. Li, Y. Zhang, X. Zhang, H. Shan, and et al., "Ct super-resolution gan constrained by the identical, residual, and cycle learning ensemble (gan-circle)," *IEEE Transactions on Medical Imaging*, vol. 39, no. 1, p. 188–203, 2020.

[109] J. Gu and J. C. Ye, "Adain-based tunable cyclegan for efficient unsupervised low-dose ct denoising," *IEEE Transactions on Computational Imaging*, vol. 7, pp. 73–85, 2021.

[110] L. Yu, M. Shiung, D. Jondal, and C. H. McCollough, "Development and validation of a practical lower-dose-simulation tool for optimizing computed tomography scan protocols," *Journal of computer assisted tomography*, vol. 36, no. 4, pp. 477–487, 2012.

[111] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.

[112] G. Toderici, D. Vincent, N. Johnston, S. Jin Hwang, D. Minnen, J. Shor, and M. Covell, "Full resolution image compression with recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5306–5314.

[113] M. Minsky and S. Papert, *Perceptrons: an introduction to computational geometry*. MIT Press, 1969.

[114] G. E. Hinton, J. L. McClelland, and D. E. Rumelhart, "Distributed representations," in *Parallel distributed processing: Explorations in the Microstructure of Cognition*. MIT Press, 1986, pp. 77–109.

[115] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[116] M. I. Jordan, *Learning in Graphical Models*. Springer, 1998.

[117] B. E. Boser, I. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of ACM Conference on Computational Learning Theory, COLT, USA*, 1992, pp. 144–152.

[118] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The journal of Physiology*, vol. 148, pp. 574–591, 1959.

[119] L. E. Atlas, T. Homma, and R. J. M. II, "An artificial neural network for spatio-temporal bipolar patterns: Application to phoneme classification," in *Neural Information Processing Systems, NIPS, Denver, CO, USA*, 1987, pp. 31–40.

[120] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[121] http://jalammar.github.io.

[122] "http://www.aboutcancer.com/brain_met_mri_2.htm."

[123] M. W. Woolrich, S. Jbabdi, B. Patenaude, M. Chappell, S. Makni, T. Behrens, C. Beckmann, M. Jenkinson, and S. M. Smith, "Bayesian analysis of neuroimaging data in FSL," *Neuroimage*, vol. 45, no. 1, pp. S173–S186, 2009.

[124] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[125] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

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

[127] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.