Image Translation and Super Resolution using Generative Adversarial Networks

A Thesis Submitted in Fulfilment of the Requirement for the Award of the Degree of

Master of Engineering

In

Electronics and Communication Engineering

Submitted By

Akanksha Sharma

Roll No: 801761003

Under Supervision of

Dr. Neeru Jindal

Assistant Professor, ECED

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY), PATIALA, PUNJAB

JULY, 2019

DECLARATION

I, Akanksha Sharma hereby declare that the work presented in this thesis entitled **"Image Translation and Super Resolution using Generative Adversarial Networks"** in fulfilment of the requirement for the award of degree of Master of Engineering (ECE) submitted at Electronics and Communication department, Thapar Institute of Engineering and Technology, Patiala is an authentic record of work carried out under supervision of **Dr. Neeru Jindal** (Assistant Professor) Electronics and Communication Department, Thapar Institute of Engineering and Technology, Patiala from 2018 to 2019.

The matter presented in this thesis has not been submitted either in part or full to any other university or institute for the award of any other degree.

Akanksha

Date: 15/7/19

Akanksha Sharma
801761003

It is certified that the above statement made by the candidate is correct to the best of my knowledge and belief.

Date: 15/7/19.

Neeru

Neeru Jindal
Assistant Professor
Electronics and Communication Engineering Department
Thapar Institute of Engineering & Technology
(Deemed to be University), Patiala, Punjab

ACKNOWLEDGEMENT

I would like to express my gratitude to my supervisor **Dr. Neeru Jindal**, Assistant Professor, Electronics and Communication Engineering Department, Thapar Institute of Engineering and Technology, Patiala who has always guided me throughout the work. I will be forever grateful to her for all the lessons she has bestowed, not only in the field of research but also for leading a successful life.

I am thankful to **Dr. Alpana Agarwal**, Head of the Department, and **Dr. Amit Mishra**, Program Coordinator, Electronics and Communication Engineering Department, Thapar Institute of Engineering and Technology, Patiala who ensured the availability of all learning facilities and infrastructure in ECED.

Akanksha Sharma

ABSTRACT

With plethora of development in the field of artificial intelligence, the use of forged images has increased manifold in deep learning algorithms. Previously, they were widely used in cyber-crimes and treachery. However, the use of forged images has gone beyond malpractices and thievery. Forged images are now produced for marketing of products, testing of algorithms, training of deep learning algorithms for self-driven vehicles and drones, etc. For generation of forged images, generative adversarial networks (GAN) are the most widely used algorithm today. Introduced by Ian Goodfellow in the year 2014, GANs have seen rapid development in the past five years. Various new models of GAN have been produced for different applications like conditional image generation, image-to-image translation, single image super-resolution, etc. With so many models GANs have been applied to different fields and expanding their scope beyond image generation.

One such application of GAN is image translation, which has proven useful in analysis of new models, translation of medical images from one domain to another and generation of new datasets. The proposed work in this dissertation is based on two applications of GAN. The first application performs image translation on disjoint and unpaired images of different bird species from CBNWI-50. CBNWI-50 is deep learning bird image dataset comprising 50 species and 5102 original images of birds commonly found in north western region of Indian subcontinent. The resolution of the translated images was improved using SR GAN which was pre-trained on CBNWI-20, DIV2k and Flickr2k datasets.

The second application was developed to obtain CT scan images of breast cancer tumours using PET scan images and vice-versa. Cross domain, unpaired images from ACRIN FLT Breast dataset were used for translation using Cycle GAN. To improve the resolution of translated images, SR GAN was used. In order to preserve the size and position of cancer tumours in the translated image, a U-Net structure was used as a feature extractor.

The simulations were carried out on python 3.6 environment using the Tensorflow and Pytorch backend on NVIDIA CUDA 9.1 graphics driver. The simulation results on both bird species and medical images were evaluated using performance parameters like PSNR, SSIM, MSE, MAE and FID score. An improvement of 12%-13% in FID score was observed on bird species in comparison to existing work. The images generated by the algorithm were also evaluated from a human perspective in order to determine their precision. However, an improvement of 5% in PSNR was achieved in medical image translation. In future, the image translation model can be further improved by using progressively growing GAN (Pro GAN).

TABLE OF CONTENTS

Sr. No.	Name of the Chapters	Page No.
	Declaration	*i*
	Acknowledgement	*ii*
	Abstract	*iii*
	Table of Contents	*v-viii*
	List of Tables	*ix*
	List of Figures	*x-xiii*
	List of Abbreviations	*xiv*
Chapter 1	**Introduction**	1-11
1.1	Preamble	1
1.2	Generative Adversarial Networks	2
1.2.1	Basic Building Blocks of Generative Adversarial Networks	3
1.2.1.1	Convolutional Neural Network	3
1.2.1.2	Back Propagation	4
1.2.1.3	Role of Loss Function	4
1.2.1.4	Generator Network	4
1.2.1.5	Discriminator Network	4
1.2.2	Loss Functions used in GANs	5
1.2.2.1	Discriminator	5
1.2.2.2	Generator	5
1.2.3	Training of Generative Adversarial Networks	6
1.2.4	Challenges in Training Generative Adversarial Networks	7
1.2.4.1	Mode Collapse	8
1.2.4.2	Vanishing Gradients	8
1.2.4.3	Non – Convergence	9
1.2.4.4	Lack of a universal performance evaluation parameter	9
1.2.5	Mitigation Methods	9
1.3	Applications of Generative Adversarial Networks	10
1.4	Dissertation Outline	11

Chapter 2 **Literature Survey** 12-22
2.1 Introduction 12
2.2 Review of Enhancement in Basic GAN Models 12
2.3 Review of Generative Adversarial Networks used for Image Translation 17
2.4 Review of Generative Adversarial Networks used for Image Resolution Improvement 19
2.5 Review of Generative Adversarial Networks used for Medical Image Translation 20
2.6 Gaps in Study 22
2.7 Research Objectives 22
2.8 Summary 22
Chapter 3 **Generative Adversarial Networks** 23-41
3.1 Convolution Neural Network 23
3.1.1 Convolution Layers 23
3.1.2 Zero Padding 24
3.1.3 Stride 24
3.1.4 Kernel 25
3.1.5 Rectified Linear Unit (ReLU) 25
3.1.6 Leaky ReLU 26
3.1.7 Pooling Layer 26
3.1.8 Batch Normalization (BN) 26
3.1.9 Instance Normalization (IN) 27
3.1.10 Sigmoid Neuron 27
3.1.11 Deconvolution Layer 28
3.2 Residual Blocks and ResNet 28
3.3 Cycle GAN 31
3.3.1 Basic Architecture 31
3.3.2 Loss Functions used in Cycle GAN 32
3.3.2.1 Adversarial Loss 32
3.3.2.2 Cycle Consistency Loss 33
3.3.3 Working 33
3.3.3.1 Discriminator Net 33
3.3.3.2 Generator Net 33

3.4 Super Resolution GAN 34
3.4.1 Loss Functions used in SR GAN 35
3.4.1.1 Content Loss 35
3.4.1.2 Adversarial Loss 35
3.4.2 Architecture and Working 36
3.4.2.1 Discriminator Net 36
3.4.2.2 Generator Net 36
3.5 U-Net 38
3.6 Selection of Performance Parameters 40
3.6.1 Mean Absolute Error 40
3.6.2 Mean Squared Error 40
3.6.3 Peak Signal to Noise Ratio 40
3.6.4 Structural Similarity Index 40
3.6.5 Frećhet Inception Distance 41
3.7 Summary 41
Chapter 4 **Results and Discussions** 42-67
4.1 Introduction 42
4.2 Datasets 43
4.2.1 Deep Learning Dataset on Common Birds of North Western India 43
4.2.1.1 Database Introduction 44
4.2.1.2 Data Collection 45
4.2.1.3 Camera Details 45
4.2.2 ACRIN FLT Breast Dataset 46
4.2.3 DIV2K Dataset 46
4.3 Geometrical Attacks Applied on Images of CBNWI 47
4.3.1 JPEG Compression and Bicubic Compression 47
4.3.2 Image Blurring 48
4.3.3 Noise Addition 48
4.3.4 Canvas Flipping and Image Rotation 48
4.3.5 Brightness and Contrast Adjustments 48
4.4 Proposed Flow Chart 52
4.4.1 Model – I (For Image Translation on Aves) 52
4.4.1.1 Image pre processing 52

4.4.1.2 Cycle GAN 52
4.4.1.3 Super Resolution GAN 52
4.4.2 Model – II (For Image translation on Breast Cancer Images) 54
4.4.2.1 Cycle GAN 54
4.4.2.2 SR GAN 54
4.4.2.3 U-Net 55
4.4.3 Machine Configuration and Environment Details 55
4.5 Results 56
4.5.1 Simulation results on Birds 56
4.5.1.1 Gender translations on same species 56
4.5.1.2 Translations on species from same genus but different species 56
4.5.1.3 Translations on species from same family but different genus and species 57
4.5.2 Simulation Results on Breast Cancer Tumour Translation 64
4.6 Summary 67
Chapter 5 **Conclusion and Future Scope** 68-69
5.1 Conclusion 68
5.2 Future Scope 69
References 70
Appendix 76
List of Publications 79

LIST OF TABLES

Sr. No.	Table Details	Page No.
Table 2.1	*Summary of major GAN base models.*	16
Table 2.2	*Summary of major GAN models which perform translation.*	18
Table 2.3	*Summary of major GAN publications which perform resolution improvement.*	20
Table 2.4	*Summary of major GAN publications in the field of medical image translation.*	21
Table 3.1	*Layer - wise description of Generator and Discriminator used in Cycle GAN.*	34
Table 3.2	*Layer - Wise description of Generator and Discriminator of SR GAN*	37
Table 3.3	*Layer - wise description of U-Net architecture*	39
Table 4.1	*Publically available deep learning datasets on Aves.*	44
Table 4.2	*Features of CBNWI-50*	44
Table 4.3	*Training parameters for Cycle GAN*	53
Table 4.4	*Training Parameters for SR GAN*	54
Table 4.5	*Machine Configuration and Environment Details*	55
Table 4.6	*Quantitative Results obtained for all the 3 simulations carried out for Stage - I and Stage – II.*	59
Table 4.7	*Comparison of Various Performance Parameters with other Publications.*	66

LIST OF FIGURES

Sr. No.	Figure Details	Page No.
Figure 1.1	*Block diagram of Generative Adversarial Network.*	2
Figure 1.2	*Basic Architecture of a Convolutional Neural Network*	3
Figure 1.3	*Comparison of the three cost functions used for generator network.*	6
Figure 1.4	*Block Diagram of steps involved in training GAN.*	7
Figure 1.5	*Depiction of Mode Collapse on MNIST dataset.*	8
Figure 1.6	*Applications of Generative Adversarial Networks*	11
Figure 2.1	*Schematic view of major GAN models. (a) Original GAN model (b) Info GAN (c) BEGAN/ EBGAN (d) Conditional GAN (e) Pix2Pix (f) Cycle GAN and (g) LapGAN. 'z' is the noise vector, c is condition labels, G is the generator and D is the discriminator. Subscript g refers to generated sample and subscript r refers to real sample.*	15
Figure 3.1	*The process of convolution using a Sobel filter and an image.*	24
Figure 3.2	*An image padded with zero intensity pixels to perform desired operations covering entire image.*	24
Figure 3.3	*Demonstration of sliding window operation, kernel and stride.*	25
Figure 3.4	*Graphical representation of rectified liner unit function.*	25
Figure 3.5	*Demonstration of max pooling operation using a 2×2 kernel on an image. Pooling also results in down-sampling of the image.*	26
Figure 3.6	*Illustration of batch normalization process.*	27
Figure 3.7	*Graphical Representation of the sigmoid function.*	27
Figure 3.8	*Illustration of the deconvolution operation on images.*	28

Figure 3.9	*Generalized architecture of ResNet.*	29
Figure 3.10	*The counter intuitive problem faced by deep networks where accuracy is less than the shallow network.*	29
Figure 3.11	*A residual block helps the deep network to learn the identity mapping by skipping the layers in between.*	30
Figure 3.12	*Two of the most common configuration of residual blocks used in deep networks.*	30
Figure 3.13	*Basic Flow Diagram of Cycle GAN*	31
Figure 3.14	*Architecture of Cycle GAN, where N denotes the number of features, K denotes the kernel or filter size and S represents the Stride.*	32
Figure 3.15	*Architecture of SR GAN, where N denotes the number of features, K denotes the kernel or filter size and S represents the Stride.*	36
Figure 3.16	*Architecture of U-Net, where N denotes the number of features, K denotes the kernel or filter size and S represents the Stride.*	39
Figure 4.1	*A collage of images from the proposed dataset, CBNWI-50.*	42
Figure 4.2	*A collage of some of species from CBNWI-50*	43
Figure 4.3	*Images from CUB-200 dataset*	46
Figure 4.4	*The first row shows PET scan images from the dataset. The second row denotes the CT scan images of the patients. It should be noted that the database is not aligned and the CT scan images shown above are not corresponding to the PET images shown above.*	47
Figure 4.5	*Sample images from CBNWI. The first, third and fifth rows contain original images, while the second, fourth and sixth rows contain compressed images.*	49
Figure 4.6	*Sample images from CBNWI. The first, third and fifth rows contain original images, while the second, fourth and sixth rows*	49

contain images blurred using average filter.

Figure 4.7 *Sample images from CBNWI. The first, third and fifth rows contain original images, while the second, fourth and sixth rows contain images with Gaussian noise.* 50

Figure 4.8 *Sample images from CBNWI. The first, third and fifth rows contain original images, while the second, fourth and sixth rows contain images which are flipped horizontally.* 50

Figure 4.9 *Sample images from CBNWI. The first, third and fifth rows contain original images, while the second, fourth and sixth rows contain images rotated by an angle of 5 degree.* 51

Figure 4.10 *Sample images from CBNWI. The first, third and fifth rows contain original images, while the second, fourth and sixth rows contain images rotated by an angle of 5 degree.* 51

Figure 4.11 *Steps for image pre-processing for model - I.* 52

Figure 4.12 *Basic flow diagram of proposed model for model - I simulation.* 53

Figure 4.13 *Pre-Processing steps followed for Model - II simulation.* 54

Figure 4.14 *Flow diagram for model - II simulation.* 55

Figure 4.15 *Translation performed for inter genus conversions (Cattle Egret to Indian Pond Heron)* 57

Figure 4.16 *Translation performed for inter species conversions (Pied Kingfisher to White Throated Kingfisher)* 58

Figure 4.17 *Translation performed for inter genus conversion reverse mapping (Indian Pond Heron to Cattle Egret)* 58

Figure 4.18 *Translation performed for inter genus conversion (Yellow Footed Green Pigeon to Laughing Dove)* 59

Figure 4.19 *Translation performed for inter species conversion reverse mapping (White Throated Kingfisher to Pied Kingfisher)* 60

Figure 4.20 *Translation performed for inter family conversion (Oriental Magpie Robin to White Browed Wagtail)* 60

Figure 4.21 *PSNR vs. Epochs plot and SSIM vs. Epochs plot for Inter - Species Translation (Pied Kingfisher to White Throated Kingfisher)* 61

Figure 4.22 *MSE vs. Epochs plot and FID vs. Epoch plot for Inter - Species Translation (Pied Kingfisher to White Throated Kingfisher)* 61

Figure 4.23 *PSNR vs. Epochs plot and SSIM vs. Epochs plot for Inter - Genus Translation (Cattle Egret → Indian Pond Heron)* 62

Figure 4.24 *MSE vs. Epochs plot and FID vs. Epoch plot for Inter - Genus Translation (Cattle Egret → Indian Pond Heron)* 62

Figure 4.25 *MSE vs. Epochs plot and FID vs. Epoch plot for Inter - Genus Translation (Cattle Egret → Indian Pond Heron)* 63

Figure 4.26 *MSE vs. Epochs plot for Inter - Species Translation (Oriental Magpie Robin → White Browed Wagtail)* 63

Figure 4.27 *Presented above are results obtained for CT → PT using the proposed model. The (a) column is the ground truth, (b) is the translated image and (c) is the recreated image using the translated image.* 64

Figure 4.28 *Presented above are results obtained for PT → CT using the proposed model. The (a) column is the ground truth, (b) is the translated image and (c) is the recreated image using the translated image.* 65

Figure 4.29 *PSNR(dB) and SSIM vs Epoch Graph for PT → CT conversion.* 65

Figure 4.30 *PSNR(dB) and SSIM vs Epoch Graph for CT → PT conversion.* 66

Figure 4.31 *MSE vs Epoch Graph for PT → CT and CT → PT conversions.* 66

LIST OF ABBREVIATIONS

BEGAN	Boundary Equilibrium Generative Adversarial Network
BN	Batch Normalization
CBNWI	Common Birds of North Western India
cGAN	Conditional Generative Adversarial Network
CIFAR	Canadian Institute For Advanced Research
CNN	Convolutional Neural Network
D	Discriminator
DCGAN	Deep Convolutional Neural Network
DISCO GAN	Discover Cross-Domain Relations with Generative Adversarial Networks
DRAGAN	Deep Regret Analytic Generative Adversarial Networks
EBGAN	Energy Based Generative Adversarial Network
ESR GAN	Enhanced Super Resolution Generative Adversarial Network
FVBN	Fully Visible Belief Nets
G	Generator
GAN	Generative Adversarial Network
HRGAN	Haze Removal Generative Adversarial Network
ICA	Independent Component Analysis
IN	Instance Normalization
Info GAN	Information Generative Adversarial Network
LSGAN	Least Squares Generative Adversarial Network
LSUN	Large-scale Scene Understanding
MCA	Markov Chain Approximations
MNIST	Modified National Institute of Standards and Technology database

MS COCO	Microsoft Common Objects in Context
RaGAN	Relativistic Averaged Generative Adversarial Network
ReLU	Rectified linear unit
RNN	Recurrent Neural Networks
RRDB	Residual-in-Residual Dense Block
SNGAN	Spectral Normalization Generative Adversarial Network
SR GAN	Super Resolution Generative Adversarial Network
SVHN	Street View House Numbers
VAE	Variational Auto-encoder
WGAN	Wasserstein Generative Adversarial Network
WGAN-GP	Wasserstein Generative Adversarial Network – Gradient Penalty

CHAPTER – 1

INTRODUCTION

1.1 PREAMBLE

Generative Adversarial Network (GAN) is derived from generative algorithms and belongs to a class of generative models along with auto-encoders (AE), fully visible belief nets (FVBN) and non-linear independent component analysis (NL-ICA). Generative models are used for approximation of training set's data distribution from latent space noise [1]. Taking generative models a step further, GAN provides the same approximation of data distribution on image datasets and generates image samples which are statistically similar to training images. GANs can be used to train models on missing data, missing data imputation, manipulation of high dimensional probability distributions and generation of multi-modal image samples [1].

Currently, the world is undergoing rapid technological development which has not only aided in comforting life of human beings but also triggered drastic climate change. Climate change [2,3] along with habitat destruction [4] and radiation [5] have started interfering with the circadian rhythm and the very existence of wildlife. Among these, birds are more susceptible to hindrance in navigation due to ambient radiation and change in weather patterns. This has resulted in dwindling of population of bird species which were once abundant, like Indian house sparrow [6]. Artificial Intelligence can play a pivotal role in conservation schemes for birds by building bird detection algorithms for aircrafts, warning systems for industries which operate in regions populated by endangered species, warning systems for common people who live in close vicinity of lakes which are populated by migratory species, educational and bird recognition apps for enthusiasts. But these tasks require training of algorithms for species recognition, detection and translation which need large number of images per species. Presently, very few datasets are available on birds and all of them lay focus on birds of North America and Europe [7-10]. Although, the existing datasets can be expanded using translation techniques but implementing the above mentioned application in a different geographical location is impossible as each locality has its own unique fauna.

Apart from flora and fauna, mankind has also become a prey of modern technology and life. Recent trends show that due to change in lifestyle and increased exposure to carcinogens in different forms has increased occurrence of cancer in human beings. Statistics show that every one in eight women might develop breast cancer at some stage of her life [11]. Recent advances in medical imaging technology have enabled mankind to detect malicious developments in the human body at an early stage. However, this boon of early diagnosis comes at an increased risk of radiation exposure during medical imaging. Various radiation-based imaging techniques like computed tomography (CT), positron emission tomography (PET), magnetic resonance imaging (MRI), X-ray, etc. are now used

widely for diagnosis of numerous ailments like cancer, various neurological disorders, cardiovascular disorders, gastrointestinal disorders, etc. But, as the precision in detection of diseases has increased, so has the radiation dose.

Solution to both of these issues can be found using generation of forged/synthetic images. Image generation is the process of creating new images with help of some prior data or description [12]. It is done so that the produced image brings an accurate representation of real scenario. When sufficient amount of training data samples are provided to neural networks, statistical methods give commendable results. However, there are certain real world scenarios in which the training data samples are not plentiful. Modern day applications like self-driving cars, automated unmanned drones, medical diagnosis databases, etc. do not have sufficient original training data. Hence, such applications may greatly benefit from forged image generation and translation techniques [13]. Forged images are used in an array of fields such as designing, advertising, film and computer generated graphic interface, simulation and training of models and algorithms, etc [14]. Among these applications, our main concentration is on image translation for the purpose of deep learning.

A useful tool for generation of images is generative adversarial net (GAN) [1]. This algorithm was introduced only five years ago, but it has gained dominance in the field of image generation in very short span of time. Due to its immense popularity in AI community, GANs have undergone a rapid development and have been implemented as task specific models other than image generation like translation [15], conditional image generation [16] and super resolution [17]. But in this dissertation the focus is aimed at image translation and resolution improvement using GAN.

1.2 GENERATIVE ADVERSARIAL NETWORKS

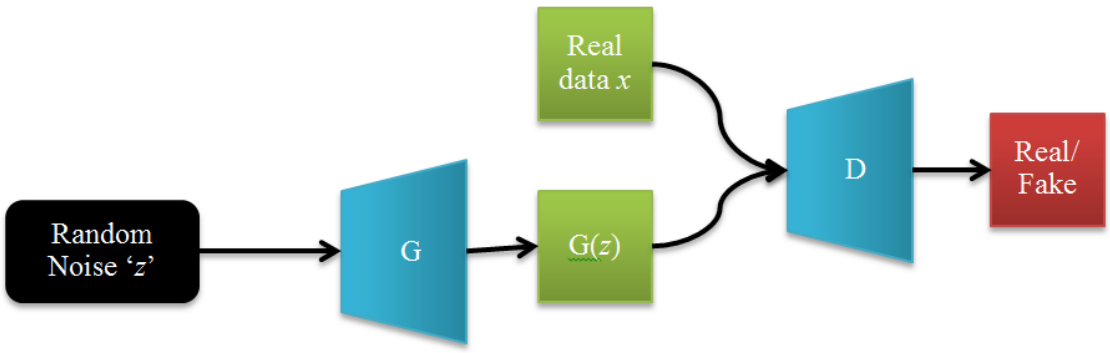


Figure 1.1 Block diagram of first Generative Adversarial Network, now referred to as Vanilla GAN. The generator network is represented as G while the discriminator network is represented as D [14].

The basic blocks of GAN are generator network (G-Net) and discriminator network (D-Net). The flow diagram of generative adversarial network is shown in Figure 1.1. Both the blocks are derived from

various variants of convolutional neural networks (CNN). While the discriminator uses images as input and returns the probability whether the image is genuine or fake, the generator takes in latent noise as input and generates an image as its output.

However, due to their massive popularity and efficiency, new GAN models have been developed. Researchers have explored various possibilities like conditional image generation through GAN (cGAN), image generation through Laplacian pyramid (LAPGAN), etc. In order to increase stability of GAN training different types of loss functions have been used like KL divergence (Vanilla GAN), JS divergence(cGAN), least squares loss (LS GSN), EM-distance (WGAN), perceptual loss and cycle consistency loss (Cycle GAN) [15, 16, 20]. All the steps involved for training vanilla GAN are described in detail in the upcoming sections.

1.2.1 Basic Building Blocks of Generative Adversarial Networks

1.2.1.1 Convolutional Neural Networks

A convolutional neural network (CNN) is based on the convolution operation and works by assigning weights to various aspects in an image. It is capable of distinguishing between various objects in an image and requires less pre-processing as compared to traditional algorithms, which require manual designing of kernels. CNN was inspired by neurons present in the visual cortex of the brain, which respond only to specific stimuli when they come in contact with a new object. The neurons in the CNN architecture are trained iteratively to learn data such that only a group or certain region of neurons responds to incoming data, by assigning weights to connections between layers [18]. This method helps the network to identify the input and produce a corresponding output. Basic working and block diagram of CNN is shown in Figure 1.2.

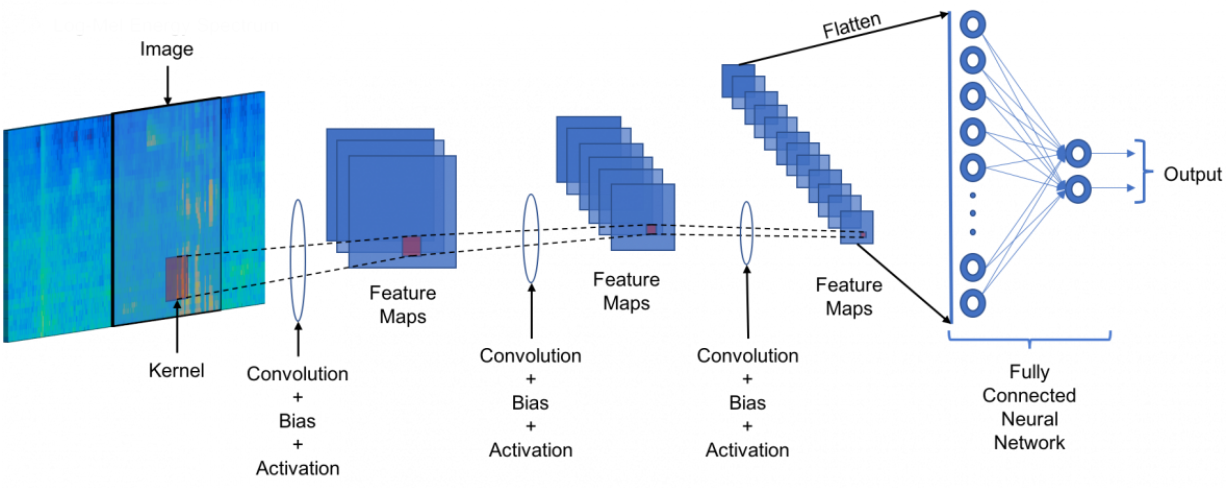


Figure 1.2 Basic architecture of a convolutional neural network [18].

1.2.1.2 Back Propagation

Back Propagation is used to calculate the gradients and weights, which are used in artificial neural networks. It works on the principle of back propagation of errors which are calculated from the output layer of the network and then redistributed throughout the network in reverse direction. It is similar to the delta rule designed for gradient computation in feed-forward and multi-layered networks. It is computed iteratively using chain rule for the calculation of gradients for each layer [19].

1.2.1.3 Role of Loss Function

Loss function evaluates how an algorithm models the input data. Loss function is usually optimized using stochastic gradient descent (SGD). If a loss function gives out a huge number as output, then it means that the predictions made by the algorithm are deviating from the target. Generally, there are two types of loss functions used in neural networks; classification loss and regression loss. Classification loss is used to predict output from a set of finite categories. Regression loss helps in prediction of continuous values from the input parameters provided to it.

1.2.1.4 Generator Network

Generator is simply embodiment of a differentiable function G in form of a deep neural network. The task of generator is to estimate the training data distribution using only the error gradient passed to it by the discriminator. Initially, the generator produces images using noise and later by the error from discriminator net. Mathematically, G(z) provides a sample of x drawn from p_{model}, when z is sampled from a prior distribution. In order to provide support to p_{model} on x space, dimensions of z should be as large as dimensions of x. The generator function should also be differentiable [20].

1.2.1.5 Discriminator Network

The discriminator is a simple classifier, which classifies the samples from input as either original or fake. This is performed by allocating probabilities to the samples. A probability of '1' means the sample is real, '0' means the sample is forged. The information in form of gradient is back propagated to generator network. This helps the generator to learn the features of the training dataset and in turn it generates images. These generated images equate to statistical properties of the original images [14].

The learning process continues as the training and generated images are fed to the discriminator network. The D-Net labels them as genuine or forged. The discriminator approximates the ratio of densities and then passes it to the generator in form of a gradient. The features are approximated jointly and alternatively by the D-Net and G-Net. In the beginning, the discriminator wins too easily, but as the training progresses the generator starts producing more realistic images. This learning of

features is largely controlled by loss functions used for training it, which are discussed in the next section [20].

1.2.2 Loss Functions used in GANs

1.2.2.1 Discriminator

The cost function of discriminator is optimized while training a binary classifier. It is trained with a sigmoid output. The classifier's training is then carried out on two mini-batches. One mini-batch hails from dataset where all samples have label '1' while the other mini-batch comes from output of generator where all the samples are labelled '0'. By training the discriminator, the ratio of where $p_{data}(x)/p_{model}(x)$ is estimated at every point x. Computation of wide variety of divergences and their gradients are enabled by this ratio. GANs base their supervised form of learning on this ratio to make approximations. GANs are also prone to failures in supervised learning. The two failures are; over-fitting and under-fitting. These failures can be overcome with perfect optimization and sufficient amount of training data [20]. The other important block of GAN is the generator whose cost functions have been stated in next section.

1.2.2.2 Generator

The generator cost functions can be implemented in different ways:

(i) Mini-max Approach

The mini-max approach results in zero-sum game which is the simplest of all versions. In this version, the sum of cost of all the players is always zero. As $J^{(G)}$ is directly associated to $J^{(D)}$, the sign can be inverted, giving us $J^{(G)} = - J^{(D)}$. The entire game can be summarized with a function which specifies discriminator's payback.

$$J^{(G)} = -J^{(D)} \tag{1.1}$$

$$V(\theta(D), \theta(G)) = -J(D)(\theta(D), \theta(G)) \tag{1.2}$$

These zero-sum games are nothing but mini-max games as their solution involves minimization on the outside and maximization on the inside.

$$\theta(G)^* = \frac{\arg}{\theta(G)} \min \frac{\max}{\theta(D)} V(\theta(D), \theta(G)) \tag{1.3}$$

As the mini-max game is amenable to theoretical analysis, it is of most interest.[20]

(i) Heuristic, Non-Saturating Approach

As the cost does not saturate even when the classifier obtains incorrect output, minimizing the entropy between the original class and the classifier's predicted class is useful. If ever the cost

saturates, i.e. tending towards zero, that means the classifier has chosen the correct class. The cross entropy minimized by discriminator is actually maximized by the generator. In early stages, the outputs generated by generator are easily rejected by discriminator. One way to prevent this, is by continuing minimization of cross-entropy by generator. Rather than truncating the sign, we change the target used to construct the cross-entropy cost [20].

$$J(G) = -\frac{1}{2} Ez \log D(G(z)) \tag{1.4}$$

(ii) Maximum Likelihood Approach

Estimating maximum likelihood with GANs results in minimization of the Kullback–Leibler (KL) divergence. It is calculated between the data and the model. Here σ is the logistic sigmoid function.

$$J(G) = -\frac{1}{2} Ez \exp(\sigma^{-1}(D(G(z)))) \tag{1.5}$$

A comparison has been made in Figure 1.3 between all the cost functions. It can be observed that the cost functions associated with mini-max approach (blue) and maximum likelihood (red) approach are tending towards zero. Both these approaches have very little gradient. This small gradient is due to the fact that if discriminator assigns a higher probability to a sample (being real), the small cost is received by the generator. Thus, we can clearly see that non-saturating heuristic approach performs the best. In the next section, training process of GAN has been described [20].

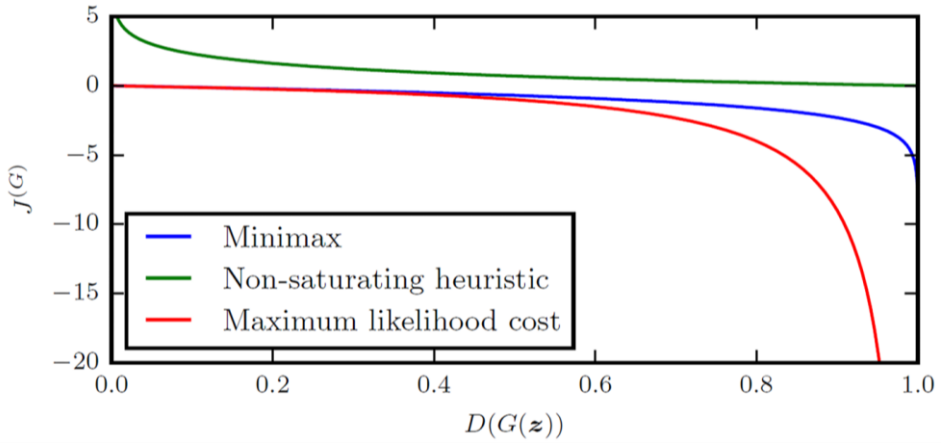


Figure 1.3 Comparison of the three cost functions used for generator network [20].

1.2.3 Training of Generative Adversarial Networks

GAN is a structured probabilistic model. The first step is to provide inputs to the generator which can be provided throughout any layer in the model and need not be the inputs to the neural network itself. G can be divided into two vectors and the first noise vector can be provided at the first layer, while the

second noise vector can be provided at the last layer. Another approach is to provide multiple random noise vectors to hidden layers of the neural net.

It consists of latent variables '*z*' (noise) and observed variables *x* (the original distribution). The G function (the generator) takes '*z*' as input and uses $\theta^{(G)}$ as parameters, while the D function (the discriminator) takes '*x*' as input and uses $\theta^{(D)}$ as parameters. Where, $\theta^{(G)}$ is the generator model and $\theta^{(D)}$ is the discriminator model. The discriminator is required to minimize $J^{(D)}(\theta^{(D)}, \theta^{(G)})$ and must do so while controlling only $\theta^{(D)}$. The generator has to minimise $J^{(G)}(\theta^{(D)}, \theta^{(G)})$ and must do so while only controlling $\theta^{(G)}$.

The solution to this optimization problem is a minimum, i.e. a point in parameter space where all other parameters have an equal or higher cost. It is solved by using Nash equilibrium, where it is a tuple $(\theta^{(D)}, \theta^{(G)})$ that is a local minima of $J^{(D)}$ with respect to $\theta^{(D)}$ and a local minima of $J^{(G)}$ with respect to $\theta^{(G)}$. By training the discriminator, an estimate is obtained for $p_{data}(x)/p_{model}(x)$ at every point in *x* domain. Computation of wide variety of divergences and their gradients are enabled by this ratio. GANs base their supervised form of learning on this ratio to make approximations. Simultaneous stochastic gradient decent is applied for training each step. Two small (mini-batches) are selected from *x* (the dataset) and the other one from *z* (model's prior over latent variables). Any of the gradient based optimization algorithm can be used to update the two gradient steps simultaneously, one updating G to reduce $J^{(G)}$ and the other updating D and reducing $J^{(D)}$. The most commonly used optimization algorithm is Adam, which is based on SGD. The training steps for GAN are summarized in Figure 1.4 [14].

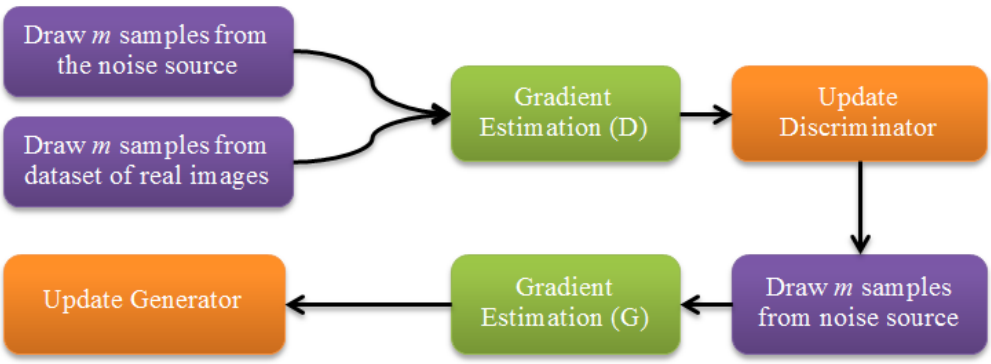


Figure 1.4 Block Diagram of steps involved in training GAN [14].

1.2.4 Challenges in Training Generative Adversarial Networks

Despite being a lucrative option to generate forged image, GANs suffer from numerous stability and performance issues, which sometimes lead to failure. These issues are usually a consequence of loss

function design or inexpedient optimization of non-convex error function. Some of the most common failures faced by GANs are elaborated in this section.

1.2.4.1 Mode collapse

Mode collapse occurs when the generator is not able to generate multiple modes present in training dataset. It is observed when multiple input noise vectors are mapped to the same point in sample space. This results in lack of diversity in the generated samples. A better way to understand this is through Figure 1.5, which shows mode collapse in MNIST dataset. The upper row indicates the generation of various digit shapes or 'modes'. The lower row depicts the case of mode collapse where the digits only indicate one mode and generate only one digit which is 6. It is a form of model under-fitting [21].

1.2.4.2 Vanishing gradients

As stated earlier, GANs operate by calculating divergence between the original model and the generated model. As the generated model improves after every step, the two data distributions start to overlap. The gradients produced by discriminator are due to non-over lapping lower dimensional manifold. The discriminator may get under the impression that it has been trained to perfection. This is a consequence of Jensen-Shanon divergence saturation and results in very small gradients or complete absence of gradients [21].

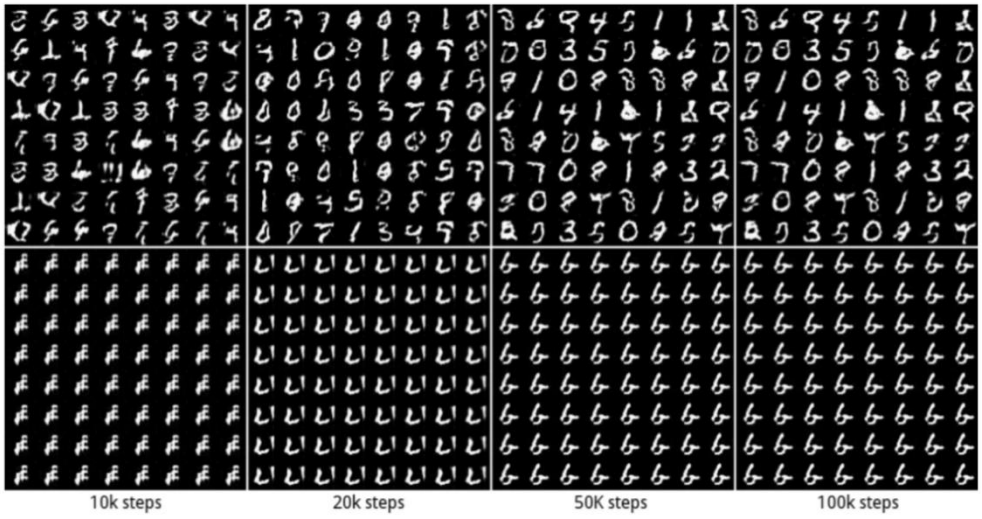


Figure 1.5 Depiction of Mode Collapse on MNIST dataset. The first row of images depicts various digits or modes while the second row shows the case of mode collapse [21].

1.2.4.3 Non-Convergence

Deciding the number of iterations while training a GAN is critical to its performance. Imbalance in iterations leads to un-damped oscillations. The convergence of a GAN model is subject to following assumptions:-

a. Infinite capacity of model
b. A convex cost function
c. Enough number of training samples

The above assumptions are not possible practically. Thus it leads to non-convergence. Although the assumptions require a convex function, yet convergence is not guaranteed even when convex functions are used [21].

1.2.4.4 Lack of a universal performance evaluation platform

The most important aspect to test any algorithm is its evaluation strategy. The ideal evaluation strategy should test all possible outcomes as well parameters. GANs suffer from lack of performance evaluation method. Several performance parameters have been used to analyse GANs, like discriminability, over-fitting detection, well defined bounds, distangled latent spaces, perceptual judgement, sensitivity to distortions and computation and sample efficiency. One evaluation metric which performs well on these parameters is the Frechet Inception Distance (FID) [22]. FID tries to find the similarities between the generated samples and the original dataset samples. Another popular metric is the inception score. It measures diversity and objectness of the produced samples. A major drawback of inception score is that it has been trained on the image-net dataset, thus it gives unreliable and even misleading results for models which have been trained on datasets other than ImageNet [22].

One of the most effective ways of judging the performance of a GAN model for realistic image forgery generation is human evaluation. A lot of GAN researchers have utilized services like AMT to verify their simulation results.

1.2.5 Mitigation Methods

The improvement techniques for GANs are motivated by heuristic understanding of non-convergence problem. Various new methods like feature matching, mini-batch discrimination, historical averaging and virtual batch normalization were proposed. Formerly GANs faced two main challenges, instability in training and mode collapse. Feature matching prevents it from overtraining on the current discriminator. It requires the generator to produce data which matches the stats of real data.

Discriminator is mostly used to specify the stats which we consider important enough to match. On an intermediate layer of discriminator, the generator is expected to match the value of features. Discriminator gets its training when it learns to discriminate between the most distinct features of real data and data that were generated by current model [21].

Another issue while training GANs is that of mode collapse. In this case, the generator collapses to a single mode and the gradient of discriminator keeps pointing to similar directions. As discriminator takes up samples one by one, it is not aware that it pointed towards a similar direction in previous case as well. Thus, all outputs start reaching a single point. This results in multiple identical outputs, about which the discriminator is not aware. This leads to non-convergence. One way to prevent this from happening is by allowing coordina9tion between samples. It can be done if the discriminator processes multiple samples in combination. This process is called mini-batch discrimination [21].

Historical averaging was inspired by fictitious play algorithm which can be used to find equilibrium. Using historical averaging, the parameters can be updated after a few iterations such that the learning rule adapts to long time series [21].

Label smoothing reduces susceptibility of GANs to adversarial examples as it changes the target value from 0.0 to 0.1 and 1 to 0.9. VBN stands for virtual batch normalization where every sample is normalized on the basis of stats collected from a reference batch [21]. Apart from Image forgery generation, GANs have various other applications which have been discussed in the next section.

1.3 APPLICATIONS OF GENERATIVE ADVERSARIAL NETWORKS

The basic purpose of GAN is generation of synthetic images. However, as various GAN models evolved to produce better quality images, researchers started exploring other areas where GANs can be used. Other than creation of forged images, GANs have been found useful in many applications. GANs are now used to generate artificial scenarios for testing of algorithms. One prime example is test of vehicular safety equipment and gear using GAN [23]. GANs have also been used for purposes like jelly fish swarm detection [24] and quality analysis of pearls [25].

Some common applications of GANs are creation of art [26, 27], single image super resolution [16], text to image generation [28], generation of videos with scene dynamics [29], Prediction of future video frame and 3-D modelling [30], predictions of missing data [31], drug discovery [32], etc. Diverse group of applications are described in the Figure 1.6 below. However, this dissertation focuses on GAN applications which deal with image translation on birds and medical images. Birds were chosen for testing the translation algorithm as they contain some of the most complex features, feather patterns and amalgam of colours. On the other hand CT scan images were used to obtain PET scan images and vice-versa using Cycle GAN.

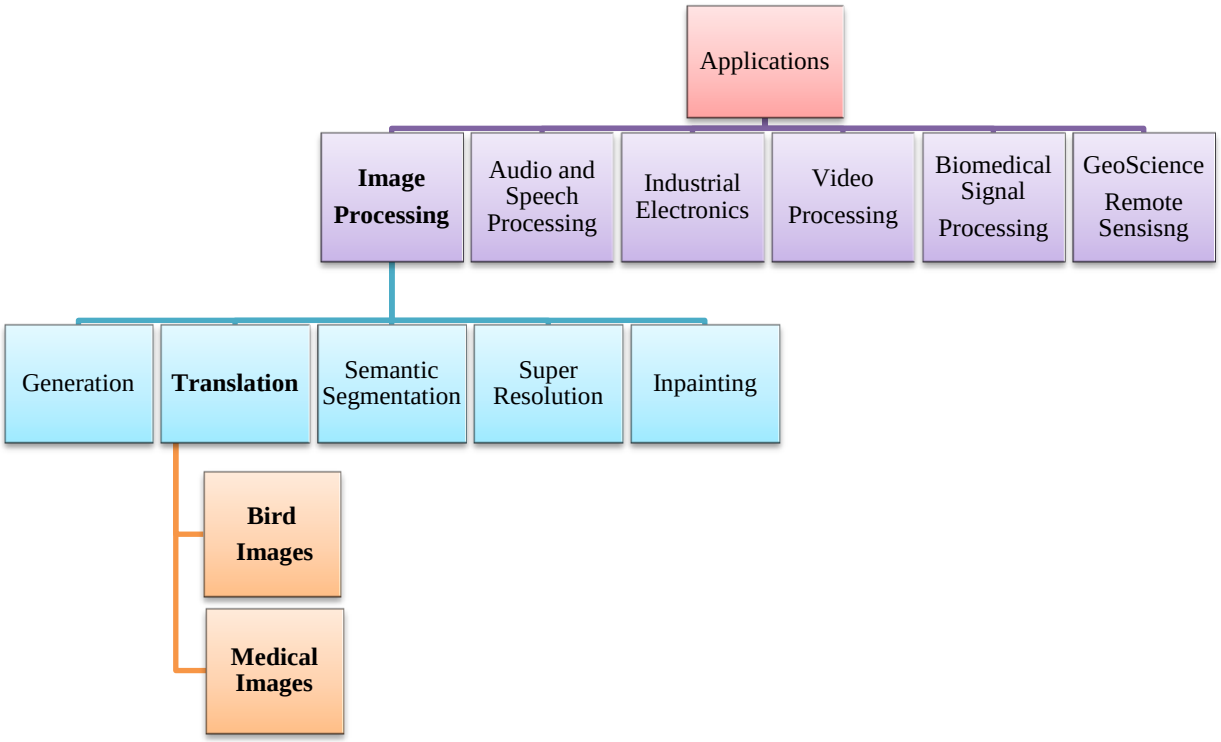


Figure 1.6 Applications of Generative Adversarial Networks

1.4 DISSERTATION OUTLINE

The dissertation aims at developing an algorithm which generates translated images of superior quality using Cycle GAN and SR GAN. The dissertation is organized into the following chapters:

Chapter 1 introduces Generative Adversarial Nets and elaborates its basic building blocks, loss functions, training process and training challenges. It also highlights mitigation methods and the applications of GAN.

Chapter 2 provides an extensive literature review on basic GAN models and enhancements made to develop application specific models. Among application specific models, algorithms built for image translation, image resolution improvement and medical image translation are reviewed in detail.

Chapter 3 illustrates the generalized structure of GAN models used in the proposed algorithm and elaborates various building blocks and working of CNN, Cycle GAN, SR GAN and U-Net. It also briefs the performance evaluation parameters used for quantitative analysis of generated images.

Chapter 4 showcases the proposed dataset and various geometrical attacks applied in detail. It also covers the explanation about both the models used and training steps. The last section of this chapter displays the result and provides their comparison with existing publications.

Chapter 5 sums up work done along with enhancements in model and dataset which can be implemented in future.

CHAPTER – 2
LITERATURE SURVEY

2.1 INTRODUCTION

In recent years, generative adversarial networks have been used in various disciplines ranging from medical image processing [33, 34], remote sensing and geosciences [35], forensic and security [36, 37], video processing [38], audio and speech processing [39], etc. This race for development of application specific GANs began after the year 2014, when Goodfellow et al. [14] built a new framework for estimating generative models using an adversarial process, where two models are trained simultaneously. Both the discriminative and generative approaches were used. The generative model learns data distribution while the discriminative model approximated the probability that whether the sample was drawn from training data or the generated data. The generator and discriminator are trained in such a way that the probability of discriminator making a mistake is maximized. This results in a two player mini-max game. When generator (G) and discriminator (D) are defined by multilayer perceptrons, the system can be trained via back-propagation.

In the first GAN model, also referred to as vanilla GAN, unrolled approximate inference networks or markov chains are not required during the training or generation of samples, which make it more efficient. With the help of qualitative and quantitative analysis of generated samples, potential of GANs is evident. It was also proposed that features from discriminator can improve performance of classifiers in semi-supervised learning, when there is a shortage of labelled data. The efficiency of training can be greatly increased by using better methods for coordination of G-net and D-net. Training can also be accelerated by determination of better distribution learning to sample z during training.

2.2 REVIEW OF ENHANCEMENTS IN BASIC GAN MODELS

GANs have undergone a rapid development in a short span of time. Few months after inception of vanilla GANs, Mirza et al. [16] introduced conditional GAN. This GAN is capable of generating samples based on class labels. The authors demonstrated these results on MNIST dataset. CGANs can also learn a multi-modal model and can be used to generate descriptive tags which are not a part of training labels. Results obtained by cGANs were not that realistic and model suffered from training instability. Another drawback of cGAN is its requirement for labelled dataset.

A generative parametric model was introduced by Denton et al. [40], called the LAPGAN, which had the capability to produce high quality specimen of natural images. Inspired from the laplacian pyramid framework, this technique was implemented using a cascade of convolution networks. The laplacian pyramid helps to generate images with rough texture and appearance, and as the pyramid progresses, finer details are added. The Gaussian pyramids are used with down sampling

characteristics. The basic idea was to break the generation into plausible successive refinements rather than any "global" notion of fidelity. It needs to be trained under supervision.

The Deep convolution generative adversarial network learns a hierarchy of representations. DCGAN [41], as it is abbreviated, is a strong candidate for unsupervised learning. But they also contain some architectural constraints. Adversarial networks learn good representations of images for supervised learning and generative modelling. To improve the stability issue in vanilla GAN, DCGAN used ReLU activation, leaky ReLU, batch normalization and fractionally strided convolution. But some forms of model instability remained, as training for longer duration resulted in oscillations. Moreover, DCGAN training parameters are not universal and need to be adjusted according to training data.

Donahue et al. [42] proposed bidirectional generative adversarial networks or BiGANs. BiGANs are capable of learning inverse mapping. The learned feature representation in BiGANs can be utilized in auxiliary supervised discrimination tasks. In some of the cases the applied network is only able to see small image sections and the global image context remains unobserved. To carry out some of the tasks, like inpainting, the image is first corrupted as large areas are removed which are to be completed by prediction network leading to inputs with very different appearance. Auxiliary information which is not available in static image domains like video, tracking or egomotion are used in some other approaches. These models are incapable of learning feature representations from unlabelled static images.

InfoGAN [43] is capable of learning detangled representations in unsupervised manner. It works on principle of maximization of mutual information. A lower bound of mutual information objectives is optimized. InfoGAN is able to discover visual concepts like hairstyles, presence and absence of eye glasses and emotions on CELEB-A face dataset. It is an unsupervised method and learns interpreTable features while adding only negligible computation costs on top of GAN. The images generated through this model lack hierarchical latent representations and using infoGAN as a high dimensional data discovery tool remains untested.

Energy based GAN [44] or EB-GAN, merged GANs and auto-encoders which resulted in a better convergence pattern and scalability. EB-GANs can generate high resolution images using single scale architecture. The discriminator is viewed as an energy function which allocates low energy to areas near data manifold and high energy to other areas, using energy as a reconstruction error. The generator here produces contrastive samples with minimal energy as opposed to the discriminator.

LS GAN introduced by Mao et al. [45] used a new loss function for its GAN model, called the least squares loss function. It is a special case of minimizing the Pearson $\chi 2$ divergence. Inspired by the issue of vanishing gradients, the authors replaced the conventional Jensen – Shanon divergence minimization. LSGAN model also worked on pulling the threshold or the decision boundary closer to the data distribution of the real data samples.

WGANs [46] used the earth-movers distance (EM distance) / Wasserstein distance for the first time and compared it with other popular methods used in deep learning. WGAN aims to minimize / optimize the EM distance. WGANs solve the major training issues in GAN. Using WGANs the network architecture and training process of GANs can be carried out easily without any worry for maintaining balance. WGANs also reduce the mode collapse issue by using weight clipping or weight pruning. Another advantage of WGANs is that it continuously estimates the EM distance, which is useful in debugging and hyper-parameter searches. The observed sample quality can also be correlated with this process. However, due to inappropriate weight clipping the WGAN model is still prone to instability, vanishing and exploding gradients gradients.

Berthelot et al. [47] proposed a new method where loss is derived from EM distance to train auto-encoder based GAN. It is a new equilibrium enforcing method which provides a new convergence method, called proportional control theory, with a robust GAN architecture. A major advantage of this method is that it balances the power between generator and discriminator while also controlling the trade-off between visual quality and image diversity. However, using an additional generative network, the auto-encoder causes the training process to slow down.

Gulrajani et al. [48] proposed improved WGAN. The authors found that weight clipping leads to difficulty in optimization. Although such problems can be solved upto a point by using batch normalization. Due to k-lipschitz constraints, WGAN models leaned towards learning extremely simple functions and ignored higher moments. The authors of improved WGAN solved this issue by using gradient penalty with accordance to lipschitz continuity. Using the k-lipschitz continuity, where 'k' is lipschitz constant, the issue of complicated optimization was resolved. Other changes made were cancelation of batch normalization for discriminator model and using the two sided penalty in order to keep 'k' close to 1. Improved WGAN proved to be much more stable for training GANs as it replaced weight clipping with gradient penalty. However, this resulted in slow convergence of model. Moreover, it also reduces the diversity of generated samples.

Mao et al. [49] improved on their previous version of LSGAN by adding gradient penalty to it. Authors made detailed comparison of the results with the non-saturating GAN models like DCGANs along with WGANs and WGAN-GP. A new performance metric was used, called the Frechet inception distance, which is based on the inception model. A detailed study was carried out to evaluate stability and image quality on different datasets. Architecture changes as well as dataset variability were explored and established the dominance of LSGANs. Although LSGAN is still prone to mode collapse, better results can be achieved if the generated samples are directly pulled towards the real data distribution.

Improved WGAN model heavily influenced researchers to come up with new normalization and penalty methods. One such model called spectral normalization GAN (SN GAN), was proposed by

Kodali et al. [50] where GAN training was studied as regret minimization, where a new gradient penalty method called Deep Regret Analytic GAN (DRAGAN), was used. Miyato et al. [51] suggested the spectral normalization technique that embodies weight normalization which has been regarded as easy to include in existing models as compared to weight normalization, weight clipping and gradient penalty.

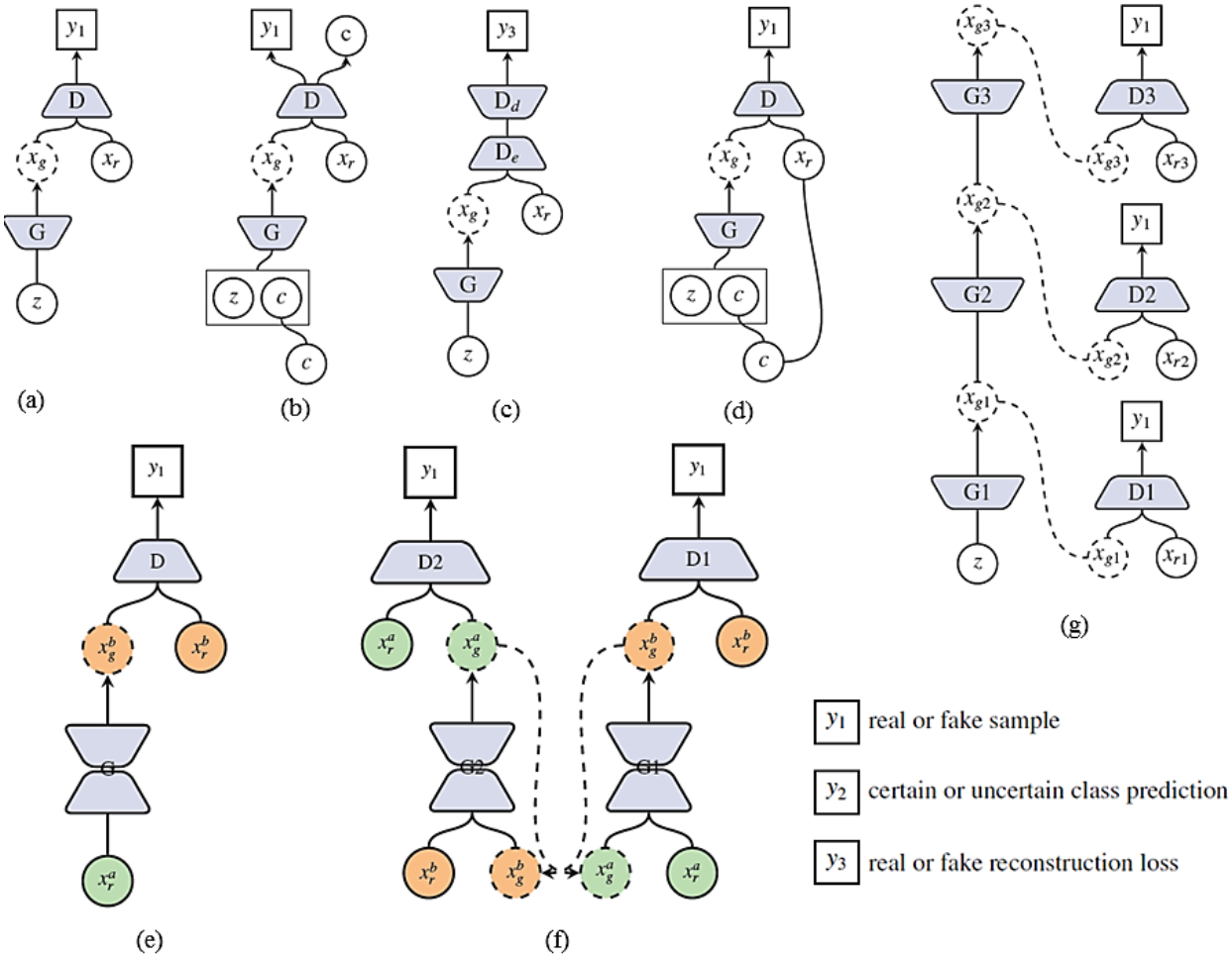


Figure 2.1 Schematic view of major GAN models. (a) Original GAN model [14] (b) Info GAN [43] (c) BEGAN/ EBGAN [44,47] (d) Conditional GAN [16] (e) Pix2Pix [56] (f) Cycle GAN [15] and (g) LapGAN [40]. 'z' is the noise vector, c is condition labels, G is the generator and D is the discriminator. Subscript g refers to generated sample and subscript r refers to real sample.

Zhang et al. [52] proposed the self-attention generative adversarial network (SAGAN). This version of GAN allows attention-driven, long range dependency modelling for image generation tasks. SAGAN can generate details in an image using cues from all feature locations unlike traditional convolutional GANs which generate details as a function of only spatially local points in low resolution feature maps. It also uses two-time up-scaling rule to speed up discriminator's training. Distant portions of the image are consistent with each other when it comes to detailed feature consistency. Spectral normalization was applied to the GAN generator as this has been established

that generator conditioning affects GAN performance. However, attention is not extended to a larger kernel size of $k \times k$.

Another improvement was proposed by Alexia Jolicoeur-Martineau [53] in form of relativistic GAN and relativistic averaged GANs. The author argued that the probability of real data being real should be decreased as the generator becomes more efficient with successive epochs. Experimental results showed that RaGANs with gradient penalty outperformed WGAN-GP and reduced the time to reach the state-of-the-art result by four times. RaGAN even produced better quality pictures from a very small set (n=2011), which is not possible with LSGAN. A preview of major GAN architectures is summarized in Figure 2.1 while Table 2.1 showcases the summary of basic GAN models.

Table 2.1 Summary of major GAN publications along with their datasets, performance parameters and contributions.

Publication	GAN Name	Speciality	Dataset Used	Performance Parameters	Summary
Goodfellow et al. [14]	Vanilla GAN	Image Generation	MNIST, Toronto Face Database, CIFAR-10	Parzen window-based log-likelihood estimates	First to introduce GANs and generate synthetic images
Mirza et al. [16]	cGAN	Conditional Image Generation	MNIST, MIR Flickr 25000	Parzen window-based log-likelihood estimates	Introduced conditional image synthesis in GANs
Denton et al. [40]	LAPGAN	Image Generation	CIFAR-10, STL, LSUN	Parzen window-based log-likelihood estimates	Used Laplacian architecture to progressively build better images, cascaded convolution, Supervised training
Radford et al. [41]	DCGAN	Image Generation	ImageNet 1k, LSUN, Faces, CIFAR-10, SVHN	Classification Accuracy	To extract more feature information, VGG network was inducted into G and D of GANs, use of ReLU, Leaky ReLU, Deconvolution, Batch Normalization
Chen et al. [43]	Info-GAN	Image Generation based on info maximization	MNIST, CelebA, SVHN, 3D Faces, 3D Chairs	Human Evaluation	Mutual information maximization, better representation of distangled spaces
Zhao et al. [44]	EBGAN	Image Generation	MNIST, LSUN, CelebA, ImageNet	IS	Use of Hinge Loss, Autoencoder as discriminator
Mao et al. [45]	LSGAN	Image Generation	LSUN, HWDB1.0, MNIST, Chinese Characters	Human Evaluation	Use of Least Squares loss for computation of divergence and quality

					improvement.
Arjovsky et al. [46]	WGAN	Image Generation	LSUN, CIFAR-10, Toy DB	IS	Use of Wasserstein Metric (Earth Movers distance)
Berthelot et al.[47]	BEGAN	Image Generation	CelebA	IS	Balances the power of the D against the G, image diversity & image quality
Gulrajani et al. [48]	Improved WGAN	Image Generation	LSUN, CIFAR-10, Toy DB	IS	Introduction of Gradient penalty to combat effects of weight clipping in WGAN model
Mao et al. [49]	Improved LSGAN	Image Generation	Cats, LSUN, MNIST,	FID	Introduction of Gradient penalty to combat effects of weight clipping in the LSGAN model
Zhang et al. [52]	Self-Attention GAN	Image Generation	ImageNet	IS, FID	Use of Self attention mechanism to obtain higher details for objects in images
Alexia Jolicoeur-Martineau [53]	RaGAN	Image Generation	Cats, CIFAR-10	FID	Use of a relativistic D and G along with spectral normalization, uses TTUR, attention limited to $k \times k$.

2.3 REVIEW OF GENERATIVE ADVERSARIAL NETWORKS USED FOR IMAGE TRANSLATION

Image to image translation is performed by learning the mapping between images using the training set of image pairs while text to image translation is done by learning mapping between text and image. Stack GAN [54] generated 256×256 photo realistic images based on text descriptions. The task was divided into two stages. Based on text description, the first stage sketches primitive shape and colours of the object providing low resolution images. The second stage utilizes results of the first stage along with the text descriptions as inputs and produces images of higher resolution. It used conditional GAN, utilizing its feature of increasing the conditioning technique which encouraged smoothness in the entire generated data. This method provided extensive quantitative and qualitative results.

Stack GAN++ [55] adopted a tree like structure with each branch containing a generator and a discriminator. It produced images at multiple scales for the same scene using different branches of the tree. When compared to Stack GAN version 1.0, the training of version 2.0 is more stable as it approximates multiple distributions. Stack GAN uses semantic image description to generate samples. Hierarchical-nested adversarial objectives help in regularization of mid-level representations. They

also help generator training to capture complex image statistics. The generated images were pushed up 17 resolutions using single stream generator architecture. This led to better adaptation of joint discriminators. Stack GAN also made use of a multipurpose adversarial loss. This encouraged effective image and text usage information usage which lead to image fidelity and semantic consistency. Also, a new method to calculate the semantic similarity of images was introduced.

The pix2pix [56] model uses the cGAN as base model, utilizing the U-Net style architecture for generator and a simple classifier as discriminator. It caused large scale reduction of parameters and produced very realistic images. However, it requires paired datasets.

Image to image translation is done by learning the mapping between images using the training set of image pairs. Zhu et al. [15] introduced a new method for translation of images from one domain to another when paired samples are missing. A cycle consistency loss was introduced to enforce this as such type of mapping is highly under-constrained. The authors also used inverse mapping. Due to cycle consistency loss in action, this GAN was named Cycle GAN, which performed the tasks of object transfiguration, photo enhancement, style transfer, season transfer, etc. Although compelling results were obtained, they were not uniform for all cases and applications.

Many failures were also observed. Cycle GAN usually succeeds when colour and texture changes are required. Although it can perform significantly better on unpaired samples too, the results are better for paired data samples. To bridge the gap between the outputs obtained via paired and unpaired data sample semantic supervision may be required. A semi-supervised training method may give power to translator algorithms and it can be much more cost efficient than fully supervised systems. The performance was not good for tasks which require geometric changes and left a scope for further improvement. Quality of images generated by Cycle GAN is lower than that produced by pix2pix. Table 2.2 showcases the summary of GAN models which perform image translation.

Table 2.2 Summary of major GAN models which perform image translation.

Publication	GAN Name	Speciality	Datasets Used	Performance Parameters	Summary
Zhu et al. [15]	Cycle GAN	Image to Image Translation	Monet, Van Gogh, Cezanne, Ukiyo-e horse to zebra Yosemite summer to winter apples 2 oranges	AMT, FCN, ppA, pcA, class IOU	Image to image translation on cross domain unpaired/ unaligned datasets, quality lower tha pix2pix
Zhang et al. [54]	StackGAN	Text to Image Translation	CUB200, Oxford102, MS COCO	IS, FID	Two stage image synthesis with two discriminators and two generators; stage I gives rough shape,

					stage II adds details
Zhang et al. [55]	StackGAN++	Text to Image Translation	CUB200, Oxford102, MS COCO	IS, FID, HR	Two stage image synthesis with multiple generators and discriminators, if stage I translation fails, stage II is also fails
Isola et al.[56]	pix2pix	Image to Image Translation	Monet, Van Gogh, Cezanne, Ukiyo-e horse to zebra Yosemite summer to winter apples 2 oranges	AMT studies, ppA, pcA, class IOU	Image to image translation on cross domain paired datasets
Yi et al. [57]	Dual GAN	Image to Image Translation	Photo-Sketch Day-Night Label-Facades Aerial Maps	AMT studies	Employs two generators and two discriminators to achieve translation, heavy computation
Kim et al. [58]	Disco GAN	Image to Image Style Transfer	CAR, FACE, Chairs, Handbag, Shoes	RMSE	Uses two GANs coupled, requires paired db, Style transfer from one domain to other, irrespective of geometric dissimilarities

2.4 REVIEW OF GENERATIVE ADVERSARIAL NETWORKS USED FOR IMAGE RESOLUTION IMPROVEMENT

Super-resolution is used to enhance the visual quality and details in an image. Ledig et al. [60] solved the problem of recovering fine texture details while super resolving at large up-scaling factors. The authors use perceptual loss which is a combination of content loss and adversarial loss. Content loss played a pivotal role in super-resolution as it takes into account the perceptual similarity rather than pixel similarity. Wang et al. [62] made significant improvements in SR GAN and introduced Enhanced SR GAN. The authors changed both the loss functions as well as the network architecture of SR GAN. They use a relativistic discriminator, similar to RaGAN. While updating the network architecture, they used Residual-in-Residual Dense Block (RRDB) as the basic building block of the network. To achieve better brightness consistency and texture recovery, ESR GAN improved perceptual loss by using features before activation. This provided better visual quality with more realistic textures. Table 2.3 showcases the summary of GAN models which perform image resolution improvement.

Table 2.3 Summary of major GAN publications which perform resolution improvement.

Publication	GAN Name	Speciality	Datasets Used	Performance Parameters	Summary
Ledig et al. [60]	SR GAN	Single Image Super Resolution	DIV2K, Set5, Set14, BSD100, BSD300	PSNR, SSIM, MOS, MSE, VGG22, VGG54	First study to achieve single image super resolution, successful but contains stray artefacts
Karras et al. [61]	ProGAN	Resolution Improvement	CelebA, LSUN, CIFAR-10	SWD, MS-SSIM	Resolution improvement using progressive growing of GANs
Wang et al. [62]	ESR GAN	Single Image Super Resolution	DIV2K, Flickr2K, OST, Set5,Set14, BSD100, Urban100, PIRM-SR	PSNR, SSIM	Enhanced SR GAN to improve hallucinated details and artefacts produced by SR GAN, use of RRDB, RaGAN, feature addition before activation

2.5 REVIEW OF GENERATIVE ADVERSARIAL NETWORKS USED FOR MEDICAL IMAGE TRANSLATION

Due to acute shortage of labelled data and medical data in general, GANs are a boon for development of synthetic medical images for automatic diagnosis, segmentation, dose calculation, etc. Using GANs can also become a cost-effective alternative to those patients who cannot afford expensive medical imaging. Some of the methods are so expensive and require heavy equipment that they are still unavailable in poor and remote areas of the world. Hence, GANs have been widely used in this field.

The first cross-domain translation in field of medical imaging using GAN was successfully carried out by Nie et al.[63]. The authors used multiple generator networks, i.e. a cascade of generators to produce realistic MRI images of brain from their counterpart CT images using the idea of auto-context model, where 3D fully connected convolutional networks are used. This simulation required paired data for CT and MRI images. Wolternik et at.[64] used Cycle GAN which does not require paired data and were able to successfully produce better results than Nie et al.. Zhao et al. [65] also performed MRI to CT conversion to carry out easier segmentation of Craniomaxillofacial Bony structure from anatomical data using deep supervision discrimination.

Chartsias et al.[66] used Cycle GAN to produce cardiac CT images using MRI, a segmentation mask from pairs of CT slices along with ground truth segmentation mask. Using a segmentation mask optimized the model performance. Jiang et al.[67] used a tumour aware loss function with Cycle GAN

to ensure that tumour is preserved during translation. Ben-cohen et al.[68] found that cGAN is useful for converting PET images to CT but less obvious tumour regions have a low response. Similarly, the authors found that FCNs are capable of producing tumours, but synthesize blurry images. Bi et al. [69] used cGAN along with label markers of tumours to synthesize more realistic output by using their multi-channel GAN, but output images have low resolution. Table 2.4 showcases the summary of GAN models which perform medical image translation.

Table 2.4 Summary of major GAN publications in the field of medical image translation.

Publication	GAN Model	Body Part	Loss	Performance Parameters
CT → MR				
Jin et. al. [70]	Cycle GAN	Brain	Adversarial, Cyclic	Segmentation
Jiang et. al [67]	Cycle GAN, Unet	Lungs	Adversarial, Cyclic, Feature, Tumour, Perceptual	Segmentation, DSC, HD95
MR → CT				
Emami et. al. [71]	cGAN	Brain	Adversarial, Image	MAE, MSE, PSNR, SSIM
Nie et. al.[63]	CascadeGAN	Brain, Abdomen	Adversarial, Image, Gradient	MAE, PSNR
MR ↔ CT				
Chartsias et. al. [66]	Cycle GAN	Cardiac	Adversarial, Cyclic	Segmentation
Wolterink et. al. [64]	Cycle GAN	Brain	Adversarial, Cyclic	MAE, MSE, PSNR
CT → PT				
Bi et. al. [69]	cGAN	Throrax	Adversarial, Image	MAE, PSNR
Ben-Cohen et. al. [68]	FCN, cGAN	Liver	Adversarial, Image	MAE, PSNR, Tumour Detection
PT → CT				
Armanious et. al. [72]	cGAN	Brain	Adversarial, Image, Perceptual, Style-content	MAE, PSNR, SSIM, VIF, UQI, FSIM
MR → PT				
Wei et. al. [73]	Casacade cGAN	Brain	Adversarial, Image	Task Specific Statistics
PT → MR				
Choi and Lee [74]	pix2pix	Brain	Adversarial, Image	SSIM, Task Specific Statistics

2.6 GAPS IN STUDY

During the literature survey, it was observed that many models of GANs have been tested on birds as they have complex features which are difficult to duplicate. But very few datasets on birds exist. Most of them are based on new world species (North and South America) or the European species. Till date, there hasn't been a dataset which provides images exclusively for the birds of Indian subcontinent.

Both basic and task specific GANs are limited to performing only one application at a time. The basic models are good at producing high quality images of lower resolution. Models which perform style transfer or image to image translation suffer from a similar problem, where output image is of poor resolution. There is a need for an algorithm which generates images rich in diversity and quality, while also keeping a decent image resolution.

After extensive survey, we observed that most attempts were made to obtain translations for brain and a considerable number for pelvis, thorax, heart and liver. But none of the translations were performed to obtain CT, PET or MRI of breast cancer tumours. Image translation in above mentioned case can be useful to prevent patient from getting exposed to higher doses of radiation, when one image modality has already been obtained. Hence, an algorithm which performs both image translation and resolution improvement on breast cancer tumour image is required.

2.7 RESEARCH OBJECTIVES

The following research objectives were identified on the basis of literature review:

1. To propose a new deep learning dataset exclusively on bird species of north western India.
2. To implement an improved algorithm for translation and resolution improvement of bird images using Cycle GAN and SR GAN.
3. To implement an improved algorithm for translation of breast cancer CT scan to PET scan and vice-versa with better resolution, while preserving the tumors in translated images.

2.8 SUMMARY

Various GAN models were studied extensively in this chapter. GANs have seen rapid development and improved manifold in a short duration. The basic vanilla GAN model has now been replaced predominantly by WGAN, cGAN and LSGAN models. These GAN variants discussed in section 2.2 are used as a base model for developing numerous task specific GANs in different fields. Further, GANs have been used in style transfer, image resolution enhancement and medical image translation which utilize the base models to develop and application. The gaps in study and research objectives identified through this comprehensive literature survey have also been summarized. The next chapter presents the constituent networks used in the proposed model in detail.

CHAPTER – 3

GENERATIVE ADVERSARIAL NETWORKS

The proposed model contains concatenated GAN algorithms for image to image translation and resolution improvement. The basic building blocks of GAN are discriminator network and generator network as discussed in chapter one. Both discriminator network and generator network are composed of convolutional neural networks. Hence to elaborate the process of image generation by GAN, convolutional neural networks, deconvolution process and other basic elements are discussed in detail. Further, the two GANs used in the model are Cycle GAN and SR GAN, which are detailed in later sections of this chapter. The last section highlights the performance evaluation methods used for quantitative analysis of generated results.

3.1 CONVOLUTION NEURAL NETWORK

3.1.1 Convolution layers

The convolution operation is performed to extract feature maps from the input images. The input to CNN is a three channel image which contains the RGB planes. Convolution is implemented by sliding the kernel over the two dimensional section of the image and calculated using dot product similar to array multiplication as shown in Figure 3.1. The result of this multiplication is a scalar called feature map, which also contains pixel information and relationship between nearby pixels. Convolution conserves the association between image pixels by learning features using small square blocks of input image.

Convolution works with the sliding window mechanism, taking the dot product between the kernel and blocks of image at all spatial locations, giving birth to local connectivity among neurons and parameter sharing. Parameter sharing is sharing of weights by all neurons in a particular feature map. The size of feature maps is also dependent on the kernel size and image size. It is impossible to use fully connected layers for high dimensional data like images. Hence, neurons are connected with only a certain region of the input image. This area is called the receptive field of the neuron. The connectivity among the neurons in the depth axis is equivalent to depth of input volume. On moving deep into convolution layers, the operation is performed between kernels and inputs of previous convolution layers. The convolution layer contains parameters which are learnable filters and results into a two dimensional activation map. The computation is faster and more efficient as the number of trainable parameters get reduced. Large number of parameters cause over-fitting of the model. The final layer or neuron of CNN depends on what function it is going to perform. CNNs usually contain fully connected layers as their second last layer and a sigmoid layer as the last layer [18].

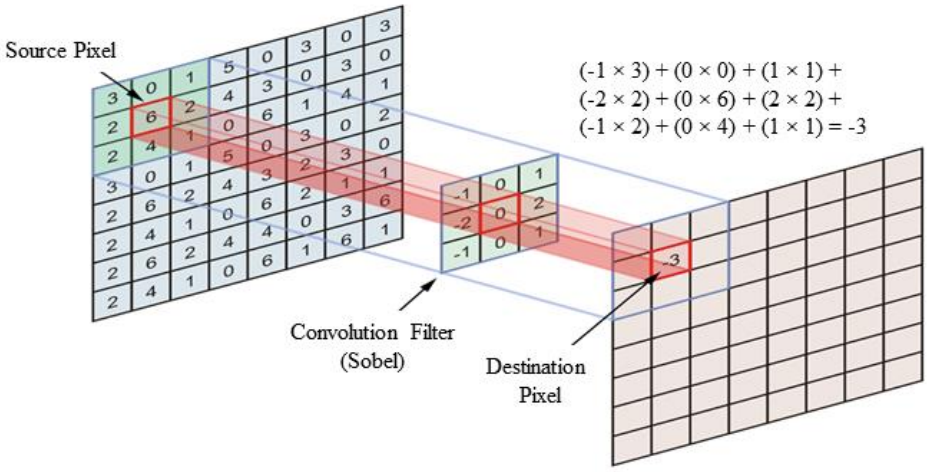


Figure 3.1 The process of convolution using a Sobel filter and an image [18].

3.1.2 Zero Padding

Zero padding is done to cover all the spatial locations of an image. Due to mismatch in image and kernel size, it is not possible to compute the convolution operation and that portion of the image needs to be dropped. To prevent this loss of information, zero padding is done. A pixel intensity value of zero (i.e. black pixels) are added to the image edges so that the kernels fit the image precisely and are able to draw more relevant feature maps. An example of zero padding is shown in Figure 3.2 [75].

Zero Padding

0	0	0	0	0	0	0
0	0	8	5	8	9	0
0	1	2	6	1	5	0
0	9	3	4	2	4	0
0	1	7	5	6	0	0
0	2	1	3	7	1	0
0	0	0	0	0	0	0

Figure 3.2 An image padded with zero intensity pixels to perform desired operations covering entire image [75].

3.1.3 Stride

Stride means step in English language. It is the step size or jump sizes of pixels executed by the kernel while the sliding window during convolution operation. If the input image is very large, higher values of strides can be used to reduce computation time and also reduces the size of feature maps. However, if stride value is increased, then it also leads to loss of fine grained information. Larger values of stride are used to extract high level feature maps. An example to demonstrate kernel, stride and sliding window operation is shown in Figure 3.3 [75].

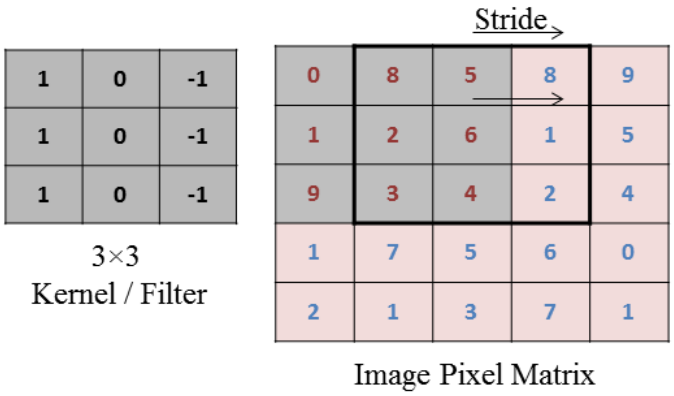


Figure 3.3 Demonstration of sliding window operation, kernel and stride [75].

3.1.4 Kernel

A kernel or a filter is preferably a square matrix which is used to extract feature maps from the input images using convolution operation. Various kernels can be used to highlight different features in an image and used for noise reduction, image enhancement, edge detection, etc. The size of the kernel decides the type, size and information captured in the feature map. A larger kernel will find and establish a relationship between more number of pixels and then leads to extraction of high level features. Usually multiple kernels are used in order to get multiple feature maps from one image. [75]

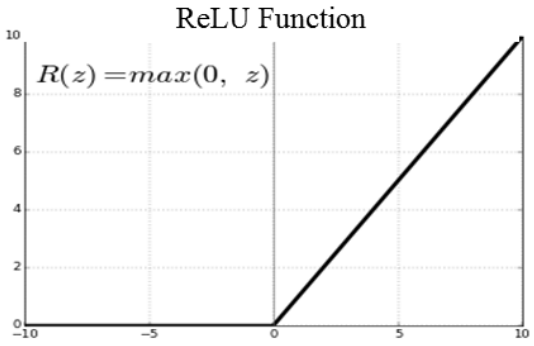


Figure 3.4 Graphical representation of rectified liner unit function [76].

3.1.5 Rectified Linear Unit (ReLU)

ReLU is rectified linear unit activation function which is provided to the neuron in networks. ReLU does not affect the size of the feature maps; instead it is used for providing threshold. The ReLU function is defined as:

$$R(z) = \max(0, z) \tag{3.1}$$

It is applied element wise for all the values in feature maps and images. It does not contain any exponentials, multiplications or divisions. Thus, it's gradient computation is easy and speeds up the training process. The negative elements obtained by error propagation are set to zero. However,

awarding zero to gradients leads to dead neurons which do not contribute to further weight updates[76]. A graphical representation of the ReLU function is shown in Figure 3.4.

3.1.6 Leaky ReLU

In Leaky ReLU, the derivative is one in the positive side and a small fraction in the negative side as opposed to zero in ReLU. Thus, leaky ReLU is not plagued by dead neurons or vanishing gradients [77].

3.1.7 Pooling layer

The pooling layer in a neural network performs down sampling of the input image along its spatial dimension. It substitutes the output of the network at selected locations with the summary statistics of nearby outputs. It moulds the extracted feature maps and makes them invariable to minute translations in the input. The most widely used pooling operation is max pooling where the maximum number from a square block of pixels or features is selected to represent it and down-sampled. As the number of pixels is reduced, more memory is freed up and computations are also faster [78]. An example of max pooling operation is illustrated in Figure 3.5.

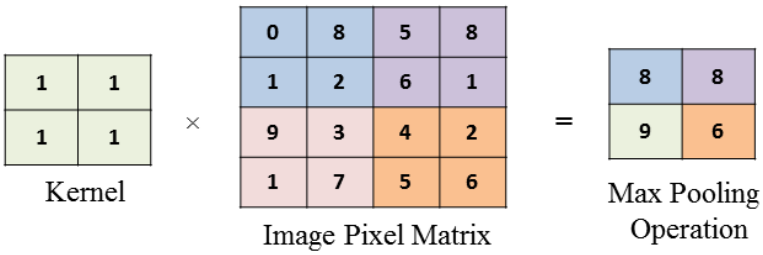


Figure 3.5 Demonstration of max pooling operation using a 2×2 kernel on an image [78].

3.1.8 Batch Normalization (BN)

The batch normalization operation is carried out to normalize output of the previous layers. It is done by computing the mean and standard deviation of the entire batch. At first the batch mean is subtracted from each instance and then each instance is divided by the batch standard deviation. This results in normalization of activations in a neural net all across the mini-batch. This process gives rise to activations having zero mean and unit standard deviation. Batch normalization is done for all features until they are exhausted. It is performed to speed up the training of the algorithm. If it is performed for the hidden layer as well, then the training speeds up drastically. It works on the principle of covariance shift, allowing the use of higher learning rate. Normalization makes sure that there is no other activation that becomes very large or very small. BN also reduces over fitting and has slight regularization effects which are somewhat similar to drop out [79]. The normalization process is illustrated in Figure 3.6.

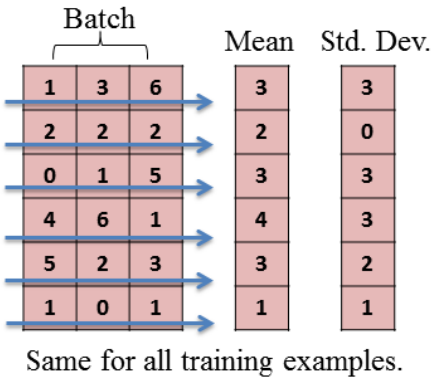


Figure 3.6 Illustration of batch normalization process [79].

3.1.9 Instance Normalization (IN)

It is possible to normalize the data on per-image basis or normalize the whole data set. Instance normalization is same as batch normalization, except for the number of input tensors that are normalized jointly. BN normalizes all images across the batch and spatial locations while IN normalizes each batch independently, i.e., across spatial locations only. BN calculates one mean and one standard deviation (thus making the distribution of the whole layer Gaussian), IN computes T of them, making each individual image distribution look Gaussian, but not jointly across each channel in each training example.[80] In GANs and style transfer applications, BN is replaced by IN.

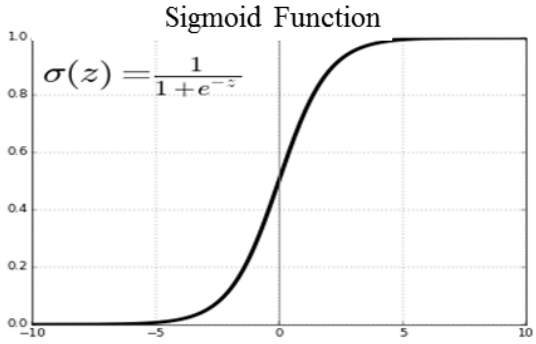


Figure 3.7 Graphical Representation of the sigmoid function.

3.1.10 Sigmoid Neuron

The sigmoid neuron is used to prevent harsh transitions faced while using the step function. Using sigmoid function does not cause a large change in output for small changes in input quantities. Sigmoid is a family of functions which have a "S" alphabet shape or "S" curve. Unlike the step function, the output obtained from a sigmoid function is a real value between 0-1 which can be interpreted as a probability. One of the most widely used sigmoid functions is the logistic function. Graphical representation of the sigmoid function is shown in Figure 3.7.

3.1.11 Deconvolution Layer

Deconvolution is a misnomer for the transposed convolution operation or the fractionally strided convolution. It helps in projecting the feature maps from a low dimensional space to a high dimensional space, while maintaining the relationship among features. It can be summarized as the gradient of convolution with respect to its input. While using deconvolution layer in python, the forward pass tries to reconstruct an image from all the principle component coefficients and the backward pass updates the principle component coefficients, provided the gradients of the image are available [81]. An example of deconvolution is shown in Figure 3.8.

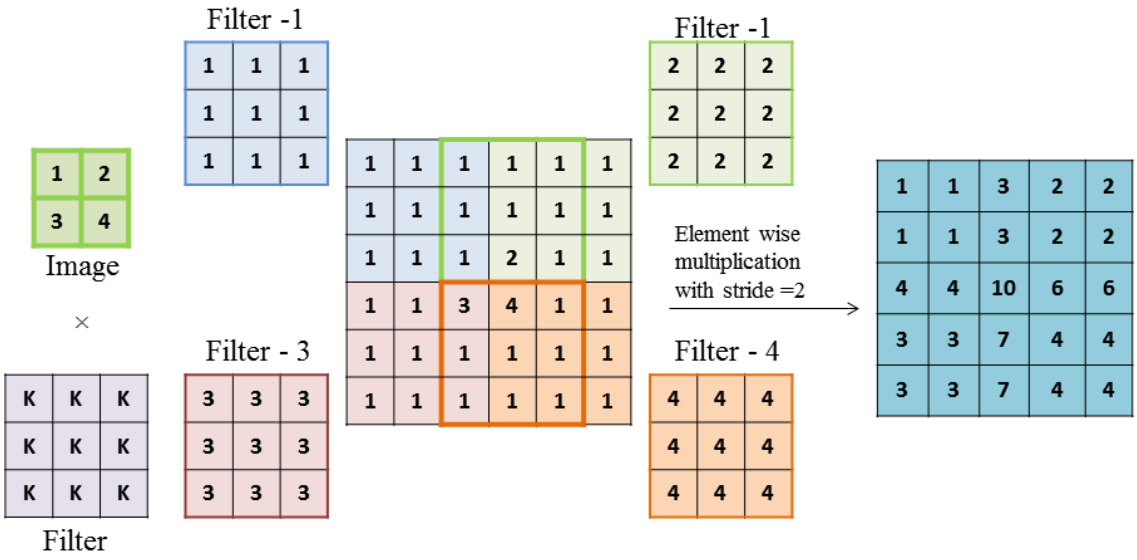


Figure 3.8 Illustration of the deconvolution operation on images. In the first step the image is padded with pixels of unit intensity and element wise multiplication is performed for each section with its corresponding filter [81].

3.2 RESIDUAL BLOCKS AND RESNET

Borrowing it's concept from universal approximation theorem, ResNet works on the principle that a feed forward network with a single layer is sufficient to represent any function. ResNet introduced an "identity shortcut connection" that skips one or more layers. ResNet is aimed at solving the accuracy degradation issue. It is a counter-intuitive issue, where accuracy in deeper networks saturates and then degrades to a very low point despite having a large number of layers. The generalized architecture of ResNet is shown in Figure 3.9.

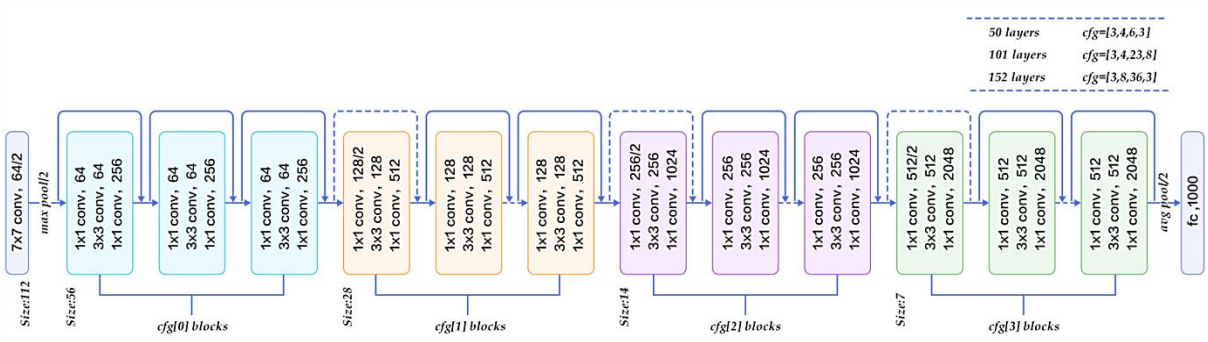


Figure 3.9 Generalized architecture of ResNet.[82]

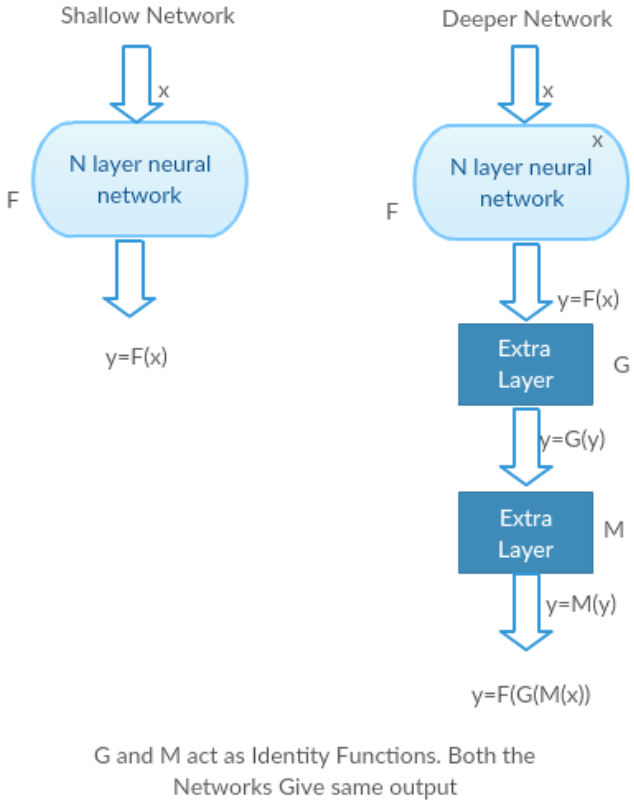


Figure 3.10 The counter intuitive problem faced by deep networks where accuracy is less than the shallow network [82].

A deep neural network *N*, having *x* layers and an average accuracy rate can be improved using state-of-the-art methods. One intuitive method to increase accuracy rate is to add some more layers and build a new Network *N'*.

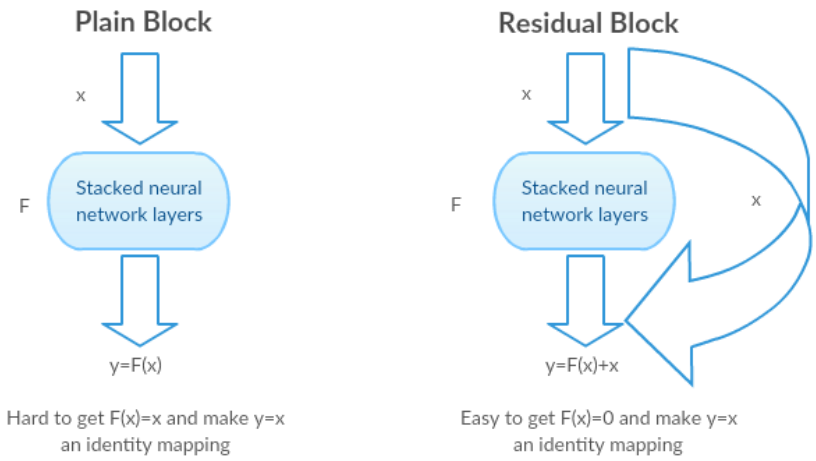


Figure 3.11 A residual block helps the deep network to learn the identity mapping by skipping the layers in between [82].

But it works under the assumption that the new network will be able to perform at least as well as the shallow network if not better. It can be established by its learning capability to learn their identity function. But experimentally it was determined that it does not happen. So, in order for the deeper network to learn the identity function, residual block is used which implements shortcuts / skip connections to enable identity function mapping. An example of identity function mapping is shown in Figure 3.10.

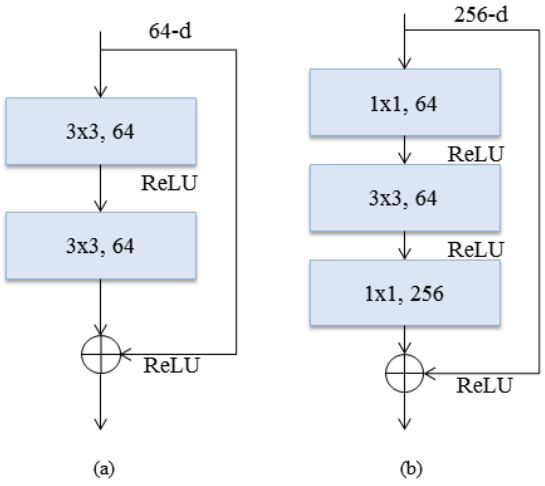


Figure 3.12 Two of the most common configuration of residual blocks used in deep networks. (a) is used in shallower networks like ResNet - 34 while (b) is used in deeper networks like ResNet - 101 and ResNet - 152 [82].

Rather than learning a direct mapping of *x* to *y* using the deep network, a residual function can be used which performs

$$H(x) = F(x) + x \tag{3.2}$$

Where, F(x) represents the identity function and x represents the stacked nonlinear layers.

This can be seen in Figure 3.11. Two layer residual blocks are used in small ResNet networks like ResNet-18 and ResNet-34. The larger three layer blocks are used in deep networks like ResNet - 50, ResNet-101 and ResNet-152. Both types of residual blocks are shown in Figure 3.12. In the upcoming sections the constituent networks used in the proposed model are discussed.

3.3 CYCLE GAN

Image to Image translation is performed by learning the mapping between images using the training set of image pairs. Zhu *et al.* [15] proposed a new method to translate images when paired samples are missing. A cycle consistency loss was used to enforce this; as such type of mapping is highly under-constrained. The authors also used inverse mapping. Due to cycle consistency loss in action, this GAN was named Cycle GAN, which performed the tasks of object transfiguration photo enhancement, style transfer, season transfer, etc.

3.3.1 Basic Architecture

Cycle GAN works on the principle of adversarial loss and cycle consistency loss. The network was proposed to carry out cross-domain unpaired image to image translations. The flow diagram of Cycle GAN is shown in Figure 3.12 and the Generator and discriminator architecture are provided Figure 3.13. The generator uses nine residual blocks derived from ResNet architecture. The discriminator is a simple convolutional neural network.

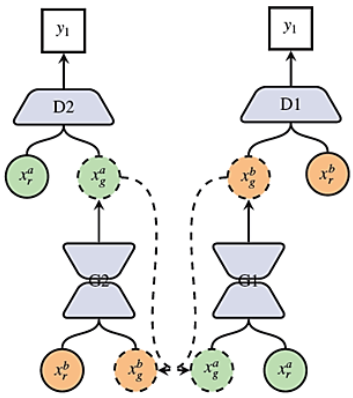


Figure 3.13 Basic flow diagram of Cycle GAN [15].

3.3.2 Loss Functions used in Cycle GAN

The Cycle GAN model uses two loss functions, namely, cyclic consistency loss and the adversarial loss.

3.3.2.1 Adversarial Loss

In this case of adversarial loss, the generator has to learn $G_{r1\rightarrow f1}$, $G_{r2\rightarrow f2}$ and fool their corresponding discriminators D_{f1} and D_{f2}.

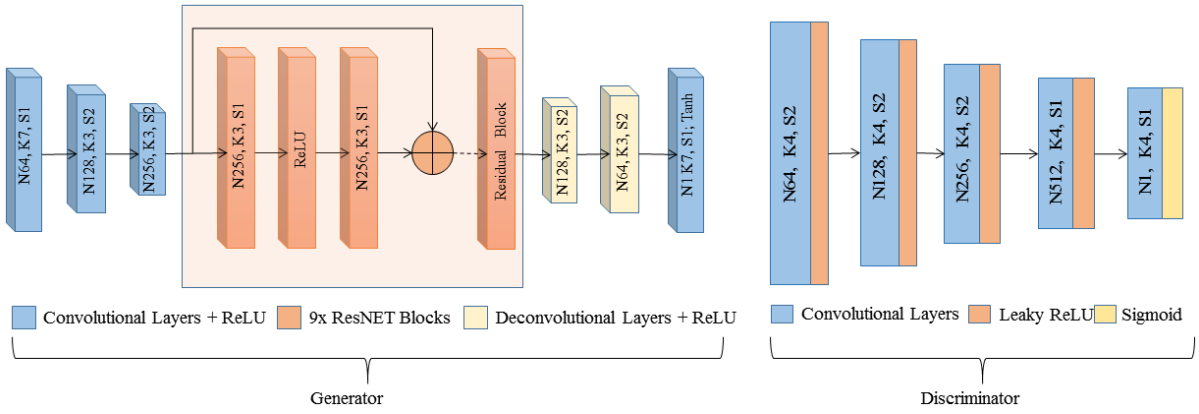


Figure 3.14 Architecture of Cycle GAN, where N denotes the number of features, K denotes the kernel or filter size and S represents the Stride [15].

Firstly, we define the adversarial loss as:

$$L_{adv}^{Fake1}(G_{r1\rightarrow f1}, D_{f1}, X_{f1}, X_{r1}) = E_{x_{f1}\sim X_{r1}}[\log(D_{f1}(x_{f1}))] + E_{x_{r1}\sim X_{r1}}[\log(1-(D_{f1}(G_{r1\rightarrow f1}(x_{r1}))))] \quad (3.3)$$

$$L_{adv}^{Fake2}(G_{r2\rightarrow f2}, D_{f2}, X_{f2}, X_{r2}) = E_{x_{f2}\sim X_{f2}}[\log(D_{f2}(x_{f2}))] + E_{x_{r2}\sim X_{r2}}[\log(1-(D_{f2}(G_{r2\rightarrow f2}(x_{r2}))))] \quad (3.4)$$

where,

$X_{r1/r2}$ and $X_{f1/f2}$ - Real and Fake images from respective image modality of type,

x_{ct} and x_{pt} - Images sampled from $X_{r1/r2}$ and $X_{f1/f2}$ domains,

$G_{r1\rightarrow f1}$ and $G_{r2\rightarrow f2}$ – Respective generators for conversion

D_{f1} and D_{f2} – Respective discriminators for conversion

Hence the adversarial losses can be summed from equation (3.3) and (3.4) as:

$$L_{adv} = L_{adv}^{Fake1} + L_{adv}^{Fake2} \tag{3.5}$$

3.3.2.2 Cycle Consistency Loss

The next loss function implemented in Cycle GAN was the cyclic consistency loss, which controls the accuracy of translations being carried between the domains. It works on the principle of cycle consistency, which means that if an image X is translated from a domain A to domain B to obtain an image Y, then the corresponding translation of image Y to domain A will yield the original image X. Cyclic consistency loss can be represented as:

$$L_{cyclic}(G_{r1\to f1,}G_{f1\to r1,}X_{r1,}X_{f1}) = E_{x_{r1}\sim X_{r1}}[\| G_{f1\to r1}(x'_{f1}) - x_{r1} \|_1] + E_{x_{f1}\sim X_{f1}}[\| G_{r1\to f1}(x'_{c}) - x_{f1} \|_1] \tag{3.6}$$

3.3.3 Working

3.3.3.1 Discriminator Net -

The discriminator net used for Cycle GAN is a simple CNN with five convolutional layers activated by leaky ReLU function and a sigmoid layer which returns a number between zero and one. The result from the sigmoid layer is considered as a probability which indicates whether the discriminator net considers them realistic or fake. Probability close to one denotes a real sample while a probability close to zero denotes a fake sample. Gradually, as the generator learns more and more features from the error back propagation, the mean probability starts to shift towards one. This probability score acts as an error which is back propagated to generator. The input to discriminator are colour images of size 128×128 which are down sampled and converted to feature maps until they reach the size of 8×8 with 512 feature channels. The last layer is the sigmoid which returns a number between 0 & 1. This number is treated as a probability which tell whether the image is fake or genuine.

3.3.3.2 Generator Net -

The generator net is composed of multiple convolution and deconvolution layers along with nine residual blocks. The first three layers in generator are convolution layers which are activated by ReLU function. These layers extract 256 feature maps in form of vectors from the image and pass them on to the residual blocks so that the network is able to learn the identity function mapping of these features. The information is passed onto the deconvolution layers which use padding and filters to generate a photo-realistic image. The layer wise description of the architecture is given in Table 3.1.

Table 3.1 Layer - wise description of Generator and Discriminator used in Cycle GAN.

GENERATOR			
Layers	**G**	**Activation**	**Input Size**
1	CONV (N64,K7,S1)	ReLU	128×128×3
2	CONV (N128, K3,S2)	ReLU	64×64×64
3	CONV (N256,K3,S2)	ReLU	32×32×128
4	RESNET BLOCK (N256, K3, S1)×9 a. Input b. CONV (N256,K3,S1),ReLU c. CONV (N256,K3,S1) (Added to layer a.)	-	32×32×256
5	DCONV (N128,K3,S2)	ReLU	32×32×128
6	DCONV (N64,K3,S2)	ReLU	64×64×64
7	CONV (N1,K7,S1)	TanH	128×128×3
DISCRIMINATOR			
Layers	**$D_{CT}(D_{PT})$**	**Activation**	**Input Size**
1	CONV (N64, K4,S2)	Leaky ReLU	128×128×1
2	CONV (N128,K4,S2)	Leaky ReLU	64×64×64
3	CONV (N256,K4,S2)	Leaky ReLU	32×32×128
4	CONV (N512,K4,S1)	Leaky ReLU	16×16×256
5	CONV(N1,K4,S1)	Sigmoid	8×8×512; Output Size - 8×8×1

3.4 SUPER RESOLUTION GAN

Super-resolution is used to enhance the visual quality and details in an image. The authors solved the problem of recovering fine texture details while super resolving at large up-scaling factors. The authors use perceptual loss which is a combination of content loss and adversarial loss. Content loss plays a pivotal role in super-resolution as it takes into account the perceptual similarity rather than pixel similarity.

3.4.1 Loss Functions used in SR GAN

The SR GAN model uses the perceptual loss function, which in turn is made up of two losses, namely content loss and adversarial loss. Content loss was derived from perceptual loss, which was specially introduced for the assessment of networks which performed style transfer and single image super resolution.

3.4.1.1 Content Loss

The content loss is based on VGG loss of the 19-layer VGG architecture, where ϕ_{ij} represents the feature map obtained after the j^{th} convolution and before the i^{th} max-pooling operation

$$L^{SR}_{VGG|i,j} = \frac{1}{W_{ij}H_{ij}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR})_{x,y})^2 \tag{3.7}$$

where,

H,W = Dimensions of resultant feature maps.

ϕ = Loss network

$G_{\theta G}$ = Gram Matrix of the style representation of the target image from the jth layer using the loss network

I^{HR}, I^{LR} = High resolution and low resolution images

3.4.1.2 Adversarial Loss

The adversarial loss can be computed as:

$$L^{SR}_{GEN} = \sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR})) \tag{3.8}$$

Hence, The combined loss for stage – II can be defined from equation (3.7) and (3.8) as:

$$L^{SR} = L^{SR}_{VGG|ij} + (0.001) \times L^{SR}_{GEN} \tag{3.9}$$

3.4.2 Architecture and Working

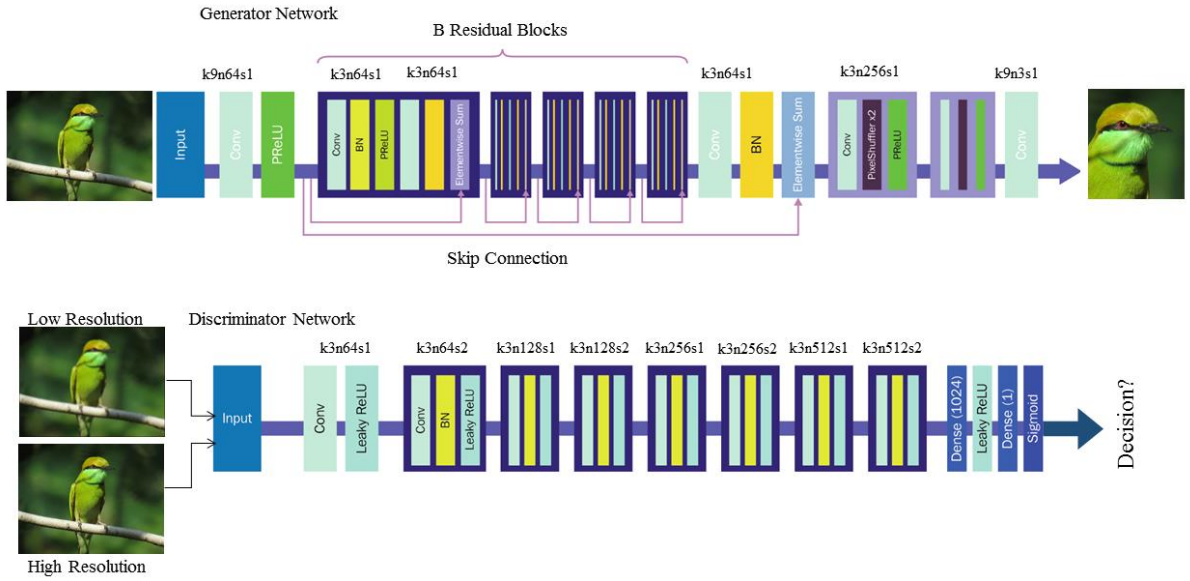


Figure 3.15 Architecture of SR GAN, where N denotes the number of features, K denotes the kernel or filter size and S represents the Stride [17].

3.4.2.1 Discriminator Net

The discriminator network is a ten layer convolutional neural network. The inputs to this network are low resolution (LR) images and high resolution (HR) images which used to train it to discriminate between HR and LR images. In SR GAN, the discriminator is trained to give lower probability to LR images and a higher probability to HR images. Multiple convolution layers with two types of strides are used to calculate perceptual loss in order to discriminate between them. At the end of the network two fully connected layers and a sigmoid function is used to provide the decision and back propagate the error to generator.

3.4.2.2 Generator Net

The generator network of SR GAN contains six residual blocks which enable it to learn feature maps of input images. Every convolution layer is followed by an activation function and a batch normalization layer. The residual blocks aid in learning the identity mapping function. The layers after the residual blocks increase the feature channels by a factor of 4× and use a pixel shuffler layer with the capacity of 2×. The pixel shuffler layer convert the pixels used up in feature channels to a spatial location in the image, doubling its size. Two pixel shuffler layers used in generator lead to an increase in size of image by a factor of 4×. If further enhancement is required another convolution layer with pixel shuffler layer can be added to improve resolution by 8×. But it leads to distortions and

empty pixel boxes and is also computationally expensive. A layer wise description of SR GAN architecture is given in Table 3.2.

Table 3.2 Layer - Wise description of Generator and Discriminator of SR GAN

GENERATOR			
Layers	**G_{SR}**	**Activation**	**Input Size**
1	CONV (N64,K9,S1)	PReLU	128×128×3
2	RESIDUAL BLOCKS (N64, K3, S1)×5 a. Input b. CONV (N64,K3,S1) c. BN d. PReLU e. CONV (N64,K3,S1) f. BN g. Element wise sum from previous	-	64×64×64
3	CONV (N64,K3,S1), BN Element wise sum from first layer	-	128×128×64
4	CONV (N256,K3,S1) Pixel Shuffler x2	PReLU	256×256×64
5	CONV (N256,K3,S1) Pixel Shuffler x2	PReLU	512×512×256 Output size - 512×512×3
DISCRIMINATOR			
Layers	**D_{SR}**	**Activation**	**Input Size**
1	CONV (N64,K3,S1)	Leaky ReLU	128×128×3
2	CONV (N64,K3,S2)	Leaky ReLU	64×64×64
3	CONV (N128,K3,S1)	Leaky ReLU	64×64×64
4	CONV (N128,K3,S2)	Leaky ReLU	32×32×128
5	CONV (N256,K3,S1)	Leaky ReLU	32×32×128
6	CONV (N256,K3,S2)	Leaky ReLU	16×16×256
7	CONV (N512,K3,S1)	Leaky ReLU	16×16×256
8	CONV (N512,K3,S2)	Leaky ReLU	8×8×512
9	DENSE(1024)	Leaky ReLU	8×8×512
10	DENSE (1)	Sigmoid	8×8×1024 Output size – 8×8×1

3. 5 U-NET

U-Net is a convolutional neural network inspired from fully convolutional network (FCN), made especially for the purpose of biomedical image segmentation. It concentrates on using data augmentation and uses the limited annotated samples more efficiently for precise localization and segmentation of medical images. The basic architecture of U-Net is shown in Figure 3.16.

U-Net can be separated into two parts, first where the feature information is maximized, called the contracting part and second where feature information is merged with spatial information, called the expansive part. The contracting part contains repetitions of convolution layers, ReLU layers and max pooling operations. The expansive part contains a large amount of feature channels which enable the propagation of contextual information to further layers. It consists of several up sampling operators associated with convolution layers, which give it u-shaped architecture.

In the process of localization, the contracting path features of higher resolution are combined with the up sampled output. A consecutive convolution layer then learns to congregate more precise output based on this information. After each down sampling layer, the number of feature channels is increased by a factor of two. Feature channels enable U-Net to distribute contextual information to higher resolution layers. The layer wise description of the architecture used is provided in Table 3.3.

In model – II of the simulations, an additional U-Net is used because it can never be guaranteed that the tumours will also be replicated while translation is carried out from one domain to another. To ensure preservation of features, U-Net is used.

From the U-Net, two types of loss, namely a feature loss and a tumour loss were derived by alternatingly training U-Net for M1 (U_{M1}) and M2 (U_{M2}).

$$L_{tumor} = E_{x_{m1}\sim X_{M1}, y_{m1}\sim y_{M1}}[\log P(y_{m1} \mid G_{M1\to M2}(x_{m1}))] + E_{x_{m1}\sim X_{M1}, y_{m1}\sim y_{M1}}[\log P(y_{m1} \mid X_{M1})] \quad (3.10)$$

The feature loss provides high level features of X_{M1} and X_{M2} as:

$$L_{feature}(x_{m1} \sim X_{M1}) = \frac{1}{C \times H \times W} \| \phi_{M1}(x_{m1}) - \phi_{M2}(G_{M1\to M2}(x_{m1})) \|^2 \quad (3.11)$$

where, ϕ_{M1}, ϕ_{M2} = High level features extracted from U_{M1} and U_{M2};

x_{m1} = Sample images of type M1

H,W = Feature size

C = Number of feature channels

$G_{M1 \rightarrow M2}$ = Generator Model for translation from M1 → M2

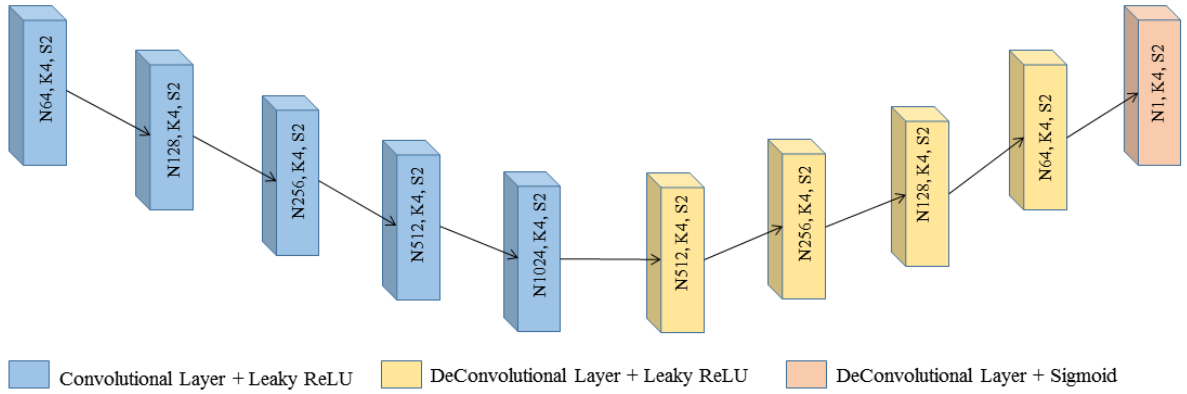


Figure 3.16 Architecture of Cycle GAN, where N denotes the number of features, K denotes the kernel or filter size and S represents the Stride [83].

Table 3.3 Layer - wise description of U-Net architecture

Layers	U-Net Layer	Activation	Input Size
1	CONV (N64,K4,S2)	Leaky ReLU	128×128×1
2	CONV (N128,K4,S2)	Leaky ReLU	64×64×64
3	CONV (N256,K4,S2)	Leaky ReLU	32×32×128
4	CONV (N512,K4,S2)	Leaky ReLU	16×16×256
5	CONV (N1024,K4,S2)	Leaky ReLU	8×8×512
6	DCONV (N512,K4,S2)	Leaky ReLU	8×8×1024
7	DCONV (N256,K4,S2)	Leaky ReLU	16×16×512
8	DCONV (N128,K4,S2)	Leaky ReLU	32×32×256
9	DCONV (N64,K4,S2)	Leaky ReLU	64×64×128
10	DCONV (N1,K4,S2)	Leaky ReLU	128×128×64; Output Size - 128×128×1

3.6 SELECTION OF PERFORMANCE PARAMETERS

GANs have numerous performance evaluation parameters hence, it was only logical to pick the most appropriate parameters according to the proposed model and application. As the proposed model uses two GANs and two different applications, it was required to choose parameters which evaluate for image quality and resolution while also enabling the proposed model to be compared with current counterparts. For resolution and image quality analysis, PSNR, SSIM and MSE were chosen. To test the diversity and objectness in generated samples, FID was selected. Thus, the quantitative evaluation was carried out using the following performance parameters:

3.6.1 Mean Absolute Error (MAE)

MAE measures the difference between the target voxel and the estimated voxel. To obtain visually similar images, it is preferred to have a smaller MAE [84]. It is calculated as follows

$$MAE = \frac{1}{N}\sum_{i=1}^{N} | I_{Syn}(i) - I_{GT}(i) | \tag{3.12}$$

3.6.2 Mean Square Error (MSE)

MSE provides the mean squared error between the target value and estimated value in an experiment. It is always a positive number and thus, sometimes, it may lead to arbitrary results. MSE is inversely proportional to PSNR (Peak Signal to Noise Ratio) [84]. Thus, as MSE decreases, PSNR of the corresponding simulation should increase.

$$MSE = \frac{1}{N}\sum_{i=1}^{N} [I_{Syn}(i) - I_{GT}(i)]^2 \tag{3.13}$$

3.6.3 Peak Signal to Noise Ratio (PSNR)

It is the ratio of maximum power of a signal to the power of noise that corrupts it. It is a standard parameter for estimation of lossy image compression codec, where higher PSNR indicates better quality [85]. For image quality estimation, it is calculated in dB by using the following mathematical relation:

$$PSNR(dB) = 10\log_{10}\frac{20^2}{MSE} \tag{3.14}$$

3.6.4 Structural Similarity Index (SSIM)

While MSE, MAE and PSNR estimate absolute errors, SSIM measures change in structural information of an image including luminance and contrast. It is measured on a scale of 0 to 1, where a higher value of SSIM indicates that the two images are visually more similar [85].

3.6.5 Frećhet Inception Distance (FID)

FID is based on the feature extraction model which is sensitive to issues like mode collapse and image diversity. It is robust towards noise in generated samples but has high bias and low variance. For calculation of Frećhet Inception Score, Inception network is used to extract features from an intermediate layer. These features are then used to model their distribution by using a multivariate Gaussian distribution with mean and covariance. The FID is calculated as:

$$FID(r,g) = \| \mu_r - \mu_g \|_2^2 + T_r(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{\frac{1}{2}}) \tag{3.15}$$

where, Tr sums up all the diagonal elements

μ_r, μ_g are the means for real and generated samples

Σ_r, Σ_g are the covariance for real and generated samples.

FID is merely the distance between training set and generated set and identifies presence or absence of features. A low FID indicates that the generated images are better in quality and have more diversity. Hence, if FID between training and testing set is calculated then it should ideally be zero, but practically FID is always non-zero [22].

3.7 SUMMARY

This chapter discusses the building blocks of both the GAN models used in this dissertation. Basic component of GANs is CNN which is used to extract features from an input image or simply provide a probability score as in the case of discriminator net. Residual blocks play a vital role in GAN generator architecture and provide a skip connection in deep neural networks for ideal mapping of features. The Cycle GAN model is used for translation and uses two types of losses, namely the adversarial loss, which is used in all types of GANs and the cyclic consistency loss which provides conversion of image from one domain to another domain. The SR GAN model uses the content loss, feature loss and adversarial loss. An additional network of U-Net is also elaborated in the text as an additional feature to work alongside the Cycle GAN model. U-Net also uses the feature loss and content loss. The performance evaluation parameters (MSE, MAE, SSIM, PSNR and FID) are discussed, which were selected to evaluate the generated images from the proposed model. The next chapter illustrates the proposed models and the simulation results.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 INTRODUCTION

Image to image translation is a deep learning task where the primary aim of the model is to learn the mapping between two target domains. It is useful where visually similar images with different feature sets are required, keeping a certain ratio of features constant. Image to image translation has proven useful in generation of new datasets, analysis and testing of new algorithms, automatic pedestrian and animal identification, conversion of medical image from one type of modality to another, etc. The simulation in this dissertation was carried out using two models, each containing two stages. In the first model, the proposed algorithm was used to carry out image translation on birds while the second model was used for performing image translation on CT and PET images of breast cancer patients.

Figure 4.1 A collage of images from the proposed dataset, CBNWI-50.

For the first model, images of birds were required. There are very few datasets on birds, thus it was limiting the scope of research. Moreover, the publically available bird datasets are mostly focused on birds of European and American descent. None of the datasets covered birds of Indian origin. Hence, to address this issue, a new database of fifty bird species from north-western India is proposed.

In first model, the target image modality and the input image modality were similar; hence the results also contained fine grained details and successfully morphed even the minute color variations. However, when the algorithm was applied to CT and PET images, the target and input image modality were dissimilar and the preservation of key features from one domain to another were

irregular. Thus, it was observed that there was still a need for an application model for the translation of breast cancer images from CT scan and PET scan.

Thus, in second model, to prevent the loss of valuable and small scale details, a U-Net was added to the algorithm, which further enhanced the performance. The role of U-Net is significant as accurate translation of tumour in the generated image is of paramount importance.

Hence, the main contributions of this dissertation are:

1. New dataset on birds of North Western India (CBNWI-50) is proposed.
2. Attacks like rotation, compression, noise addition have been used.
3. Image translation on birds is carried out by Cycle GAN and SR GAN.
4. Medical Image translation on CT and PET images of breast cancer tumours to obtain cross domain unpaired image to image translation is done using Cycle GAN, SR GAN and U-Net.
5. U-Net is used to ensure the preservation of tumours in translated images.

4.2 DATASETS

4.2.1 Deep Learning Dataset on Common Birds of North Western India

Generative adversarial networks require a large number of images to train upon. Thus, large datasets are required. The complexity of dataset also plays a deciding factor for judging an algorithm's performance. Most of the datasets are accompanied with additional information called the annotations. This is complementary information which aids the algorithms to enhance their performance by providing additional parameters to train upon. However, creating annotations for an image dataset is a tedious and grueling manual labor for the researches.

Figure 4.2 A collage of some of species from CBNWI-50

These datasets contain small pictures with varying sizes and shapes, different illumination, colors, exposure, field of depth, etc. Deep learning datasets require a large number of images with varying diversity and categories to test the efficiency of algorithms. Diversity among the samples within the datasets is vital. Datasets built for animal and bird species often have fewer number of sample images. If the species is rare, it might have very few sample images of poor quality, which are insufficient for deep learning models. Popular publicly available bird datasets are listed in Table 4.1.

Table 4.1 Publically available deep learning datasets on Aves.

Name	**No. of Species**	**Total Number of Images**	**Annotations/ Bounding Boxes**
CUB-200 [7]	200	11,788	Yes
iNaturalist [8]	964	214,295	Yes
Birdsnap [9]	500	49,829	Yes
NA Birds [10]	400	48,000	Yes

4.2.1.1 Database Introduction

During the literature survey on Ave database, it was observed that very few such datasets exist. Most of them are based on new world species (North and South America) or the European species. Till date, there hasn't been a dataset which provides images exclusively for the birds of Indian subcontinent. All the four datasets mentioned in Table 4.1 cover only the continents of North America (New World species) and Europe. Sample images from one of the most popular datasets, CUB-200 are shown in Figure 4.3. In order to build a dataset, data was collected in form of photographs of common birds found in North-Western India (predominantly Rajasthan). Table 4.2 lists the features of the proposed dataset. The entire list of species along with their respective details has been shared in appendix.

Table 4.2 Features of CBNWI-50

Camera used	**Canon Powershot SX60 HS (16 MP)**
Photograph Format	RAW, JPEG
States Included	Rajasthan, Haryana, Punjab
Total Number of Species	50
Total Number of Original Images	5,102
Labeled Images	Species and attack-wise labelling on all images
Annotations/BB	No

Total Number of Images after applying Attacks – 35,714		
Attacks	**Type**	**Parameters**
Compression	Bicubic Compression	Image Size - 300×400
Noise Addition	Gaussian Noise	μ=0, σ = 0.009
Image Blurring	Average Filter	Filter Size = 5×5
Rotation	Counter Clock Wise (CCW)	5 Degree
Contrast and Brightness Adjustment	-	Lower Bound = 0.01 Upper Bound = 0.90
Canvas Flipping	Horizontal Canvas Flip	-

The dataset contains images of 50 bird species including local as well as migratory birds which can be easily sited in the states of Rajasthan, Haryana and Punjab. Majority of the data has been collected within the 150 km radius of Jaipur City between 2016 and 2019.

4.2.1.2 Data Collection

For collection of data, locations with abundant bird fauna were identified. Birds are usually cited in marshes, shallow waters, river sides, lakes and in vegetation away from city lights and noise pollution. The most appropriate time to locate and document birds is during dawn and dusk. As India falls in temperate and tropical zones, it is visited by migratory birds from Europe and other Asian countries like China and Russia during winter season. These populations of migratory birds are also spotted in the lakes and marshes of Northern regions of the country. The data collection points were Thapar University Campus (Patiala), Sukhna Lake (Chandigarh) University of Rajasthan Campus (Jaipur), Chandlai Dam (Jaipur), Smriti Van (Jaipur), Barkhera Jain Teerth Lake (Jaipur), Jhalana Forest Reserve (Jaipur), Man Sagar Lake (Jaipur), Ana Sagar Lake (Ajmer), Gatolav Lake (Dausa) and Tilyaar Lake (Rohtak).

4.2.1.3 Camera Details

The images were photographed using Canon Powershot SX60 HS. It is a bridge camera with SLR like body by Canon. It has a high zoom range of 65x optical zoom and 4x digital zoom. This particular model was selected as it can shoot photographs in both RAW file format and JPEG file format. The sensor used in this camera is BSI-CMOS with a size of 1/2.3" (6.17mm x 4.55mm). The focal length, maximum shutter speed and aperture range are 21-1365mm, 1/2000sec and f3.4-f6.5 respectively.

Figure 4.3 Images from CUB-200 dataset [6].

4.2.2 ACRIN FLT Breast Dataset

The ACRIN – FLT – Breast dataset was provided by the courtesy of cancer imaging archive. The dataset was collected from multiple centres to establish the changes recorded by using Flourothymidine F-8 (FLT), which is a cancer PET imaging tracer. It documented proliferation during NAC (Neo – adjuvant chemotherapy) with pCR (pathologic complete response) in locally advanced breast cancer patients. It contains images of 83 breast cancer patients. The image modalities in the dataset were CT scan, PET scan and MR, in DIACOM format. For PET scans, the patients were injected with a radioactive fluoride molecule, bonded with glucose molecule. Cancer cells are known to have high metabolism and consume a higher amount to glucose to compensate that. As the cancer cells consume more and more glucose, more radioactive fluoride molecules accumulate on active sites and appear as bright 'hot spots' in PET images. The images in the dataset are longitudinal cross-sectional slices of longer PT and CT Scans of patients from multiple health centres, who participated in pre-treatment, during and post-treatment/surgery for imaging sessions. Some of the sample images from dataset are shown in Figure 4.4 [86,87].

4.2.3 DIV2K Dataset

The DIVerse 2k dataset was exclusively built for the single image super resolution challenge using royalty free and non- copyrighted images available on the internet. It contains one thousand images which have 2k resolution in at least one axis. The dataset contains a large variety of images like

people, environments, flora, fauna, still objects and monuments. The dataset was randomly split into eight hundred train, one hundred test and one hundred validation images [88].

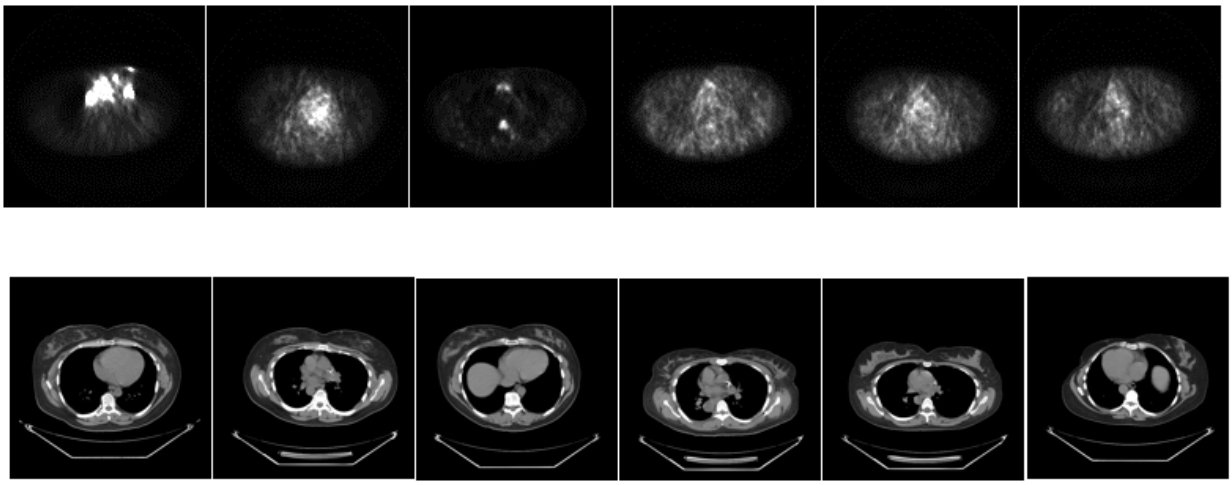


Figure 4.4 The first row shows PET scan images from the dataset. The second row denotes the CT scan images of the patients. It should be noted that the database is not aligned and the CT scan images shown above are not corresponding to the PET images shown above [86,87].

4.3 GEOMETRICAL ATTACKS APPLIED ON IMAGES OF CBNWI

The images from the proposed data set were subjected to multiple attacks in order to replicate the noise distortions, different perspectives, compression, various camera generated and pixel irregularities. Some of the two dimensional geometric attacks like rotation, scaling and canvas flipping are also an effective tool to increase the diversity of the sample images in the data set.

4.3.1 JPEG Compression and Bicubic Compression

Originally the data was collected in RAW format. It is acknowledged as RAW because such type of images are taken straight from camera and not subjected to any alterations like white balance, hue, saturation correction, etc. RAW is also referred to as a digital negative. It is advantages to use RAW file format s because it contains all the data recorded by the camera sensor along with the meta data. Such a privilege is not enjoyed while capturing the image in in JPEG format, which is a lossy compression file format. Different camera production companies use their own raw file formats. Nikon cameras used .NEF and Canon uses .CR2. RAW has greater tonal range as it contains 12 bit data while jpg contains only 8 bits of data. However, RAW images require more storage space and take long time when used in image processing and deep learning applications. Thus, at the price of small amount of information loss while keeping relevant information intact, JPEG fares better for deep learning applications and it is memory efficient. The images were converted to JPEG according to ISO standards.

The database contains 5,102 high resolution JPEG files. Deep learning requires smaller file size for quicker processing and bulky high resolution images are not desirable. Hence, the JPEG images were

compressed using bicubic compression to size 300×400. A collage presenting the uncompressed and the compressed images are shown in Figure 4.5.

4.3.2 Image Blurring

Image blurring was carried out using average filter or the mean filter. The average filter replaces value of each pixel by the mean of its neighbours, including itself, according to the filter size. Average filtering results in elimination of pixels which are peculiar to their surroundings. As a consequence, high frequency components like edges are removed and images appear blurred or smoothened. Mean filtering is similar to convolution operation which is applied using a square kernel of size 3×3 or 5×5. Large kernel size results in greater amount of smoothening effect. It is a spatial filter used for noise reduction in image processing. The filtering operation was performed using the fspecial command and imfilter command in MATLAB. A collage showcasing the blurred images along with the original images are shown in Figure.

4.3.3 Noise Addition

Noise is added to images to test the efficiency of the algorithm, to increase the variety of image samples and to duplicate the effect of images recorded in low light conditions. Principal sources of Gaussian noise in digital images arise during acquisition e.g. sensor noise caused by poor illumination and/or high temperature, and/or transmission e.g. electronic circuit noise.

Each image contains noise caused due to sensor issues, disparity in photon approximation and due to error during transmission of data. In the following simulation, Gaussian noise of mean $\mu=0$, and standard deviation of $\sigma = 0.009$ was added to the images using the imnoise command in MATLAB. A collage shows noisy images along with original images in the Figure 4.7.

4.3.4 Canvas Flipping and Image Rotation

Both image rotation and canvas flipping belong to the two dimensional geometric operations called the affine transforms. The rotation operation is performed with respect to origin of the image by mapping position of the elements of the input image to the output image, rotated at a specified angle. Canvas flipping was carried out by horizontally flipping the image and providing a different perspective of the same object. Images were rotated using the imrotate command in MATLAB by an angle of 5°. Canvas flipping was obtained using flip command in MATLAB. Collage depicting both image rotation and canvas flipping attacks are shown in Figure 4.8 and Figure 4.9 respectively.

4.3.5 Brightness and Contrast Adjustments

An image must have the proper brightness and contrast for easy viewing. Brightness refers to the overall lightness or darkness of the image. Contrast is the difference in brightness between objects or regions. The brightness and contrast in the images were adjusted to provide the dataset with data samples with varying light and exposure. This attack not only affects the background but also changes the actual colors of the birds. The brightness and contrast were adjusted using MATLAB command

imadjust with a lower bound of 0.01 and an upper bound of 0.90. The sample images are shown in Figure 4.10.

Figure 4.5 Sample images from CBNWI. The first, third and fifth rows contain original images, while the second, fourth and sixth rows contain compressed images.

Figure 4.6 Sample images from CBNWI. The first, third and fifth rows contain original images, while the second, fourth and sixth rows contain images blurred using average filter.

Figure 4.7 Sample images from CBNWI. The first, third and fifth rows contain original images, while the second, fourth and sixth rows contain images with Gaussian noise.

Figure 4.8 Sample images from CBNWI. The first, third and fifth rows contain original images, while the second, fourth and sixth rows contain images which are flipped horizontally.

Figure 4.9 Sample images from CBNWI. The first, third and fifth rows contain original images, while the second, fourth and sixth rows contain images rotated by an angle of 5 degree.

Figure 4.10 Sample images from CBNWI. The first, third and fifth rows contain original images, while the second, fourth and sixth rows contain images which have been adjusted for contrast and brightness.

4.4 PROPOSED FLOW CHART

4.4.1 Model – I (For Image translation on Aves)

4.4.1.1 Image pre processing

Image pre-processing is a crucial step while preparing a dataset to train on a deep learning algorithm. Datasets in their original form may not be suiTable or insufficient. In this simulation, the bird photographs were captured in different environments and vary greatly in terms of illumination, contrast and area covered by the object in the image. To prepare images to feed into Cycle GAN network, following pre-processing steps were carried out as shown in Figure 4.11.

1. Brightness and contrast adjustments.
2. Image cropping to ensure that the object occupies at least 40% of the image.
3. Resizing of the images to a size of 143×143.
4. Splitting the training and testing set using 0.8 and 0.2 respectively.

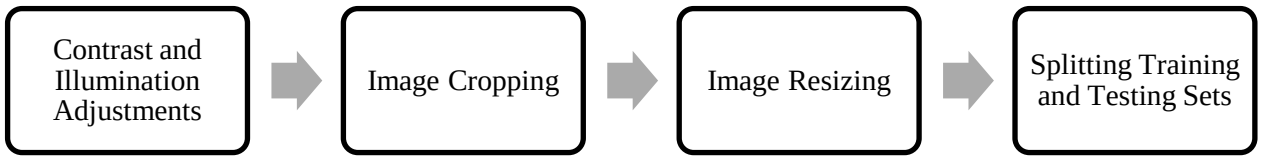


Figure 4.11 Steps for image pre-processing for model - I.

4.4.1.2 Cycle GAN

The images from CBNWI-50 were pre-processed, converted to JPEG and resized to a size of 143×143 using bicubic compression. For species which had less number of images, open source images were used from Flickr to expand the training dataset. The Cycle GAN model uses ADAM optimizer with an initial learning rate of $2e^{-4}$. The learning rate is slowly declined to zero after 100 epochs. The training is carried out for a default number of epochs (200) but can be halted if convincing results are obtained earlier than expected. The training parameters of Cycle GAN are listed in Table 4.3.

4.4.1.3 Super Resolution GAN

To train SR GAN model, DIV2K dataset was used. This dataset contains 800 training images and 100 test images of 2K size. As super resolution on birds was to be performed, an additional 500 images of birds of 2K resolution were jumbled with DIV2K dataset images to fine-tune the SR GAN model to produce more texture on bird features like feather patterns and eyes. The additional 500 bird images

were taken from CBNWI-50. SR GAN learns by using a set of low resolution (LR) and a set of high resolution (HR) images. The training parameters for SR GAN are listed in Table 4.4.

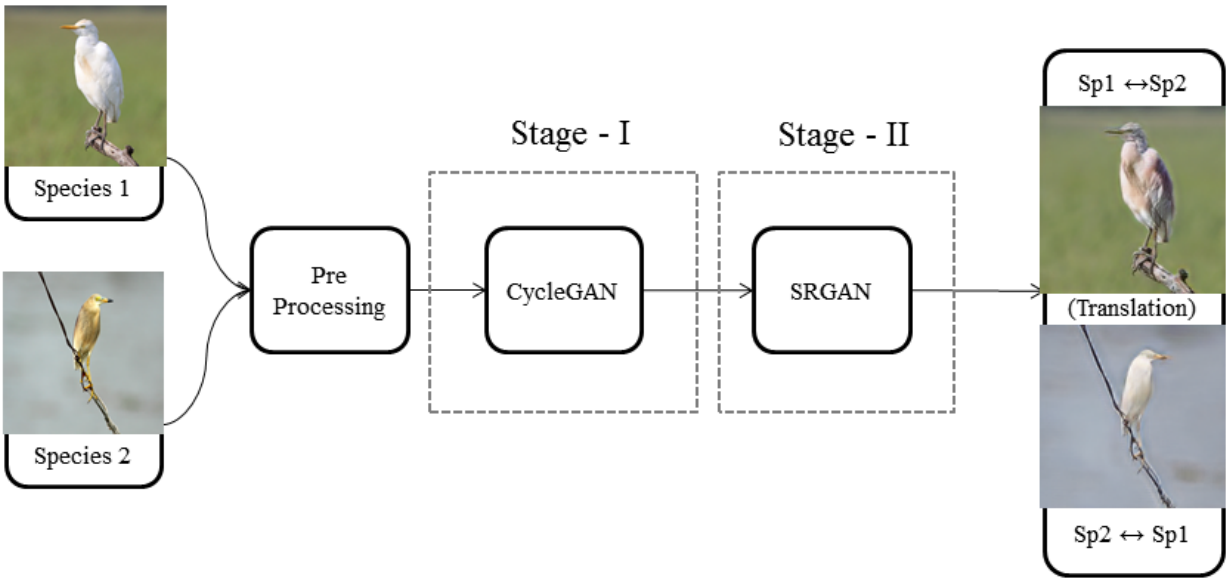


Figure 4.12 Basic flow diagram for model - I simulation.

Table 4.3 Training parameters for Cycle GAN

Batch Size	: 1	**Learning Rate**	: 0.0002
Maximum Epochs	: 200	**Learning Rate Decay Policy**	: Decay with each epoch after epoch 50.
Dataset Type	: Unaligned	**Learning Rate Decay Rate**	: 0.00001
Discriminator Architecture	: Vanilla GAN Discriminator	**Max Dataset limit**	: inf
Generator Architecture	: ResNet 9 Block Generator	**Input Image Size**	: 143×143×3
Optimizer	: ADAM	**Cropped Image Size**	: 128×128×3
ADAM Momentum (β)	: 0.5	**Max Output Image Size**	:128×128×3

Table 4.4 Training Parameters for SR GAN

Batch Size	: 16	**Learning Rate Decay**	: 1%
Learning Rate	: 0.0001	**Decay Learning Rate**	: After every 2 Epochs
Optimizer	: ADAM	**Input Image Size to G**	: 128×128×3
ADAM Momentum (β)	: 0.9	**Output Image Size**	: 512×512×3
Training Epochs	: 2000	**Scaling Factors**	: 2x,4x

4.4.2 Model – II (For Image translation on Breast Cancer Images)

The image pre-processing steps are similar to model – I, however, images were not cropped this time as the PET and CT images were already captured with the breast cancer tumour in focus. The steps for image pre-processing are shown in Figure 4.13 and the flow diagram for model – II is shown in Figure 4.14.

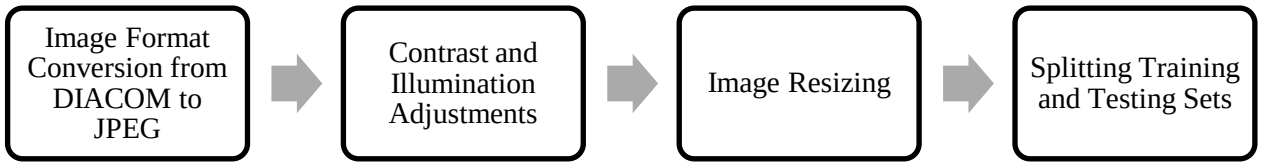


Figure 4.13 Pre-Processing steps followed for Model - II simulation.

4.4.2.1 Cycle GAN

To train Cycle GAN in this model, the DIACOM (Digital Imaging and Communication in Medicine) images from ACRIN - FLT - Breast dataset were converted to JPEG image format and resized to 143x143 using bicubic image compression in MATLAB. The remaining parameters were kept the same as model - I, listed in Table 4.3.

4.4.2.2 Super Resolution GAN

To train the SR GAN model for model II, images from the ACRIN - FLT - Breast dataset used. High resolution pictures of both the image modalities, i.e. PET scan and CT scan were used to train SR GAN. Once the training was completed, separate models for single image super resolution for CT and

PET were saved and were later utilized to obtain higher resolution images from the images generated by Cycle GAN. Training parameters for model – II were same as that of model – I, listed in Table 4.4.

4.4.2.3 U-Net

In Model - II, an additional circuit called the U-Net was added to the circuit so that the tumors in the translated images remain preserved. U-Net was developed as a medical image feature extractor and serves the purpose of extracting features from both the original dataset and the images generated using that dataset. During training, the original CT scan images and the generated PET Scan images were fed to the U-Net separately.

Later, both the feature vectors were added and again fed to Cycle GAN for further processing. The first feature vector contains the features related to size and location in input image modality (CT scan) and the second feature vector contains the relative position and size of the tumor on generated image modality (PET scan).

When this information is passed into Cycle GAN, It uses its conditional image generation network to generate images with preserved tumors. Although cyclic consistency loss is efficient in learning the mapping between the two domains, it can never guarantee that the tumours will also be replicated while translation is carried out from one domain to another. To ensure preservation of features, U-Net is used.

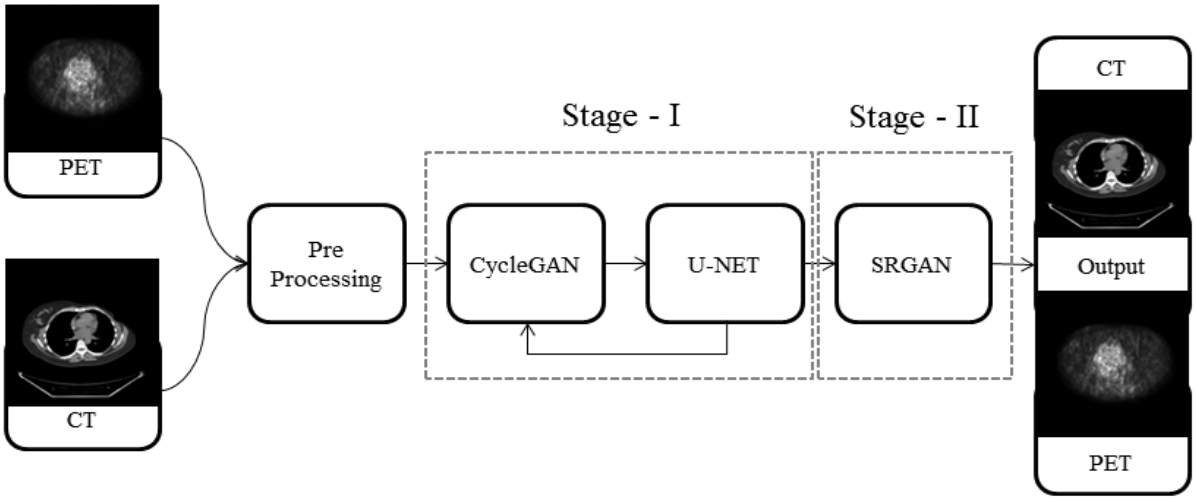


Figure 4.14 Flow diagram for model - II simulation.

4.4.3 Machine Configuration and Environment Details

Table 4.5 Machine Configuration and Environment Details

Machine Type	: MSI GRD2 Gaming Laptop	**Environment**	: Python
CPU	: i7 – 8th Generation Intel CPU	**Python Version**	: 3.6

CPU RAM	: 16 GB	**Backends**	: Pytorch , Tensorflow
Operating System	: Windows 10 Pro	**Software Used**	: Anaconda, Spyder, MATLAB
GPU	: NVIDIA GeForce GTX 1050Ti	**GPU RAM**	: 4 GB

4.5 RESULTS

4.5.1 Simulation results on Birds

In order to explore the limits to which style transfer can be performed in birds, three different types of simulations on different bird species were carried out based on their scientific species classification. Scientific species classification is described as following; Kingdom (Animalia) > Phylum (Chordata) > Class (Aves) > Order > Family > Genus > Species. First experiment was performed for inter-species translations, second for inter-genus translations and third for inter-family translations. Inter-Order translations were also attempted, but they were not as successful as other translations due to vast morphological differences between birds of different orders. The translations are shown in Figure 4.15 to Figure 4.20.

Human perception based evaluation is one of the best performance metric for translation tasks. It can be clearly observed that the translation of one bird species to another was successful in most cases. However, it was noted that translated images for closely related bird species are much better when compared to inter -family and inter-order translations.

4.5.1.1 Gender translations on same species

Translations can also be performed when male and female of the same species. However, it does not apply to those species where, male and female are of completely different size and features, like a peacock and a peahen (*Pavo cristatus*).

4.5.1.2 Translations on species from same genus but different species

Genus is a categorization in scientific classification which lies above species level and below family level. Animals and plants which are phylogenically related are classified under same genus. Better translations are observed when translations are made on species of same genus, like in the case of white throated Kingfisher (*Halcyon smyrnensis*) and pied kingfisher (*Halcyon rudis*) (Figure 4.16 and Figure 4.19).

4.5.1.3 Translations on species from same family but different genus and species

Family is a taxonomic classification level below order and above genus. Members of a family level classification share a common trait or attribute in their appearance. Some inter-family translations were very successful, like Cattle Egret (*Bubulcus ibis*) (Figure 4.15) and Indian Pond Heron (*Ardeola grayii*) (Figure 4.17). Similar translation was also carried out between laughing dove (*Spilopelia senegalensis*) and yellow footed green pigeon (*Treron phoenicoptera*) (Figure 4.18)

When transition is made to broader classification terms like different families and different orders, the build and feather patterns start to vary greatly. In such circumstances, the translations are not as successful as in lower classification categories. One such example is translation between oriental magpie robin (Order: Passeriformes Family: Muscicapidae Genus: *Copsychus* Species: *saularis*) to white browed wagtail (Order: Passeriformes Family: Motacillidae Genus: *Motacilla* Species: *maderaspatensis*) (Figure 4.20).

The quantitative results for this analysis have been listed in Table 3. The best results have been highlighted in all the segments. Figure 4.21 – 4.26 showcase various quantitative analysis graphs obtained for all three simulations.

Figure 4.15 Translation performed for inter genus conversions (Cattle Egret to Indian Pond Heron)

Figure 4.16 Translation performed for inter species conversions (Pied Kingfisher to White Throated Kingfisher)

Figure 4.17 Translation performed for inter genus conversion reverse mapping (Indian Pond Heron to Cattle Egret)

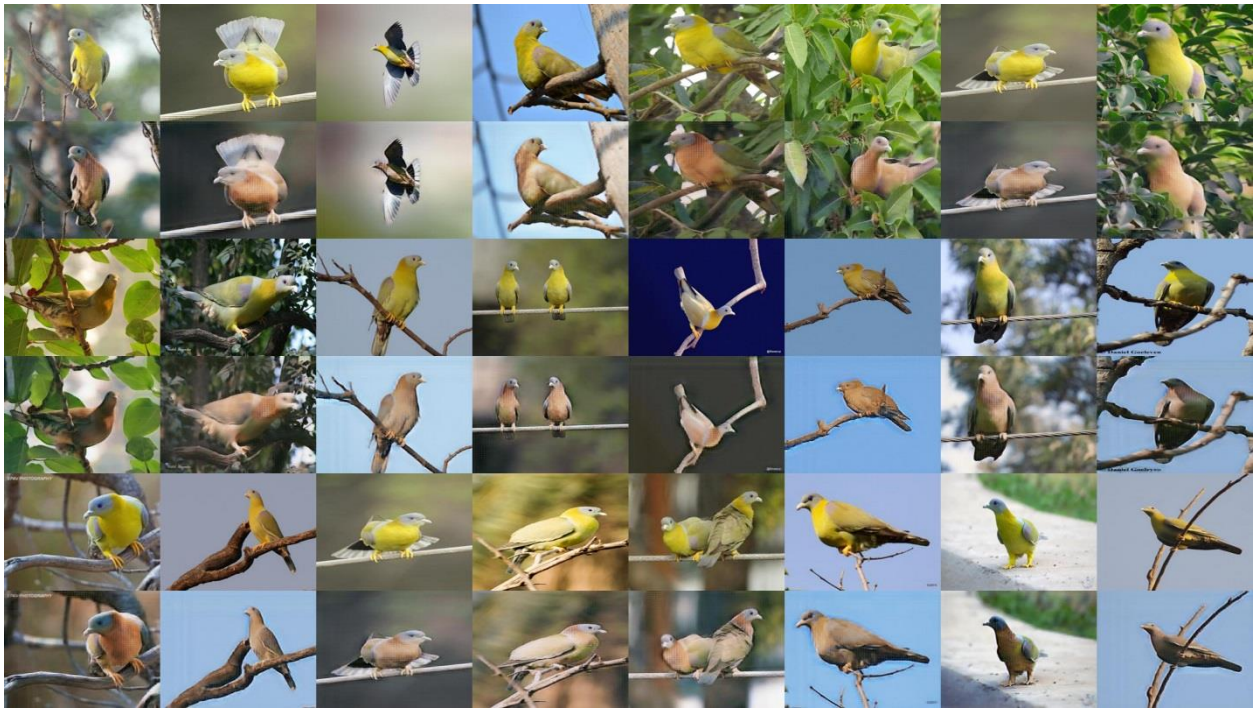


Figure 4.18 Translation performed for inter genus conversion (Yellow Footed Green Pigeon to Laughing Dove)

Table 4.6 Quantitative Results obtained for all the 3 simulations carried out for Stage - I and Stage – II. The results are compared with Stack GAN which performed style transfer on CUB dataset. * indicates that the images were resized to 64*64 before computation of FID.

Translation	**Family**	**Genus**	**Species**	**PSNR (dB)**	**SSIM**	**MSE**	**FID**
Stage – I							
Inter – Species	Same	Same	Different	30.5926	0.9679	56.7309	-
Inter – Genus	Same	Different	Different	32.2363	0.9592	38.8550	-
Inter – Family	Different	Different	Different	29.2560	0.9566	77.1763	-
Stage – II							
Inter – Species	Same	Same	Different	**33.5918**	**09887**	30.1926	**38.11**
Inter – Genus	Same	Different	Different	33.5855	0.9779	**28.4795**	54..12
Inter – Family	Different	Different	Different	32.8863	0.9730	33.4541	78.79
Zhang et al. [54]	-	-	-	-	-	-	51.89
Zhang et al.*[55]	-	-	-	-	-	-	35.11

Figure 4.19 Translation performed for inter species conversion reverse mapping (White Throated Kingfisher to Pied Kingfisher)

Figure 4.20 Translation performed for inter family conversion (Oriental Magpie Robin to White Browed Wagtail)

Quantitative Analysis Graphs for Inter – Species Translations

(Pied Kingfisher to White Throated Kingfisher)

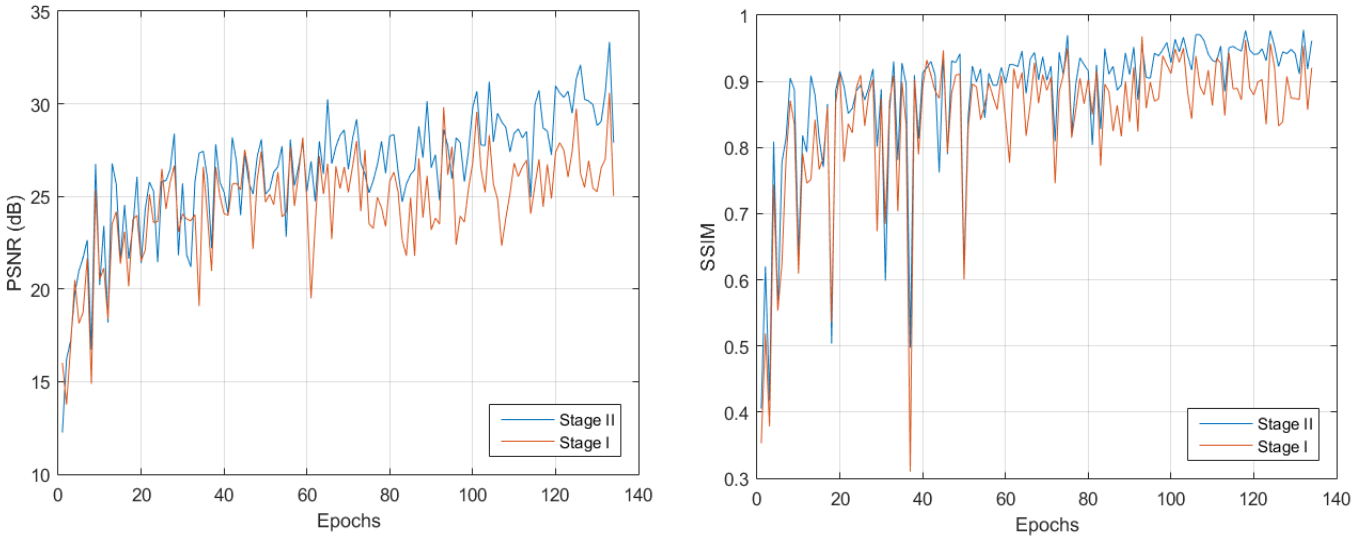


Figure 4.21 PSNR vs. Epochs plot and SSIM vs. Epochs plot for Inter - Species Translation (Pied Kingfisher to White Throated Kingfisher)

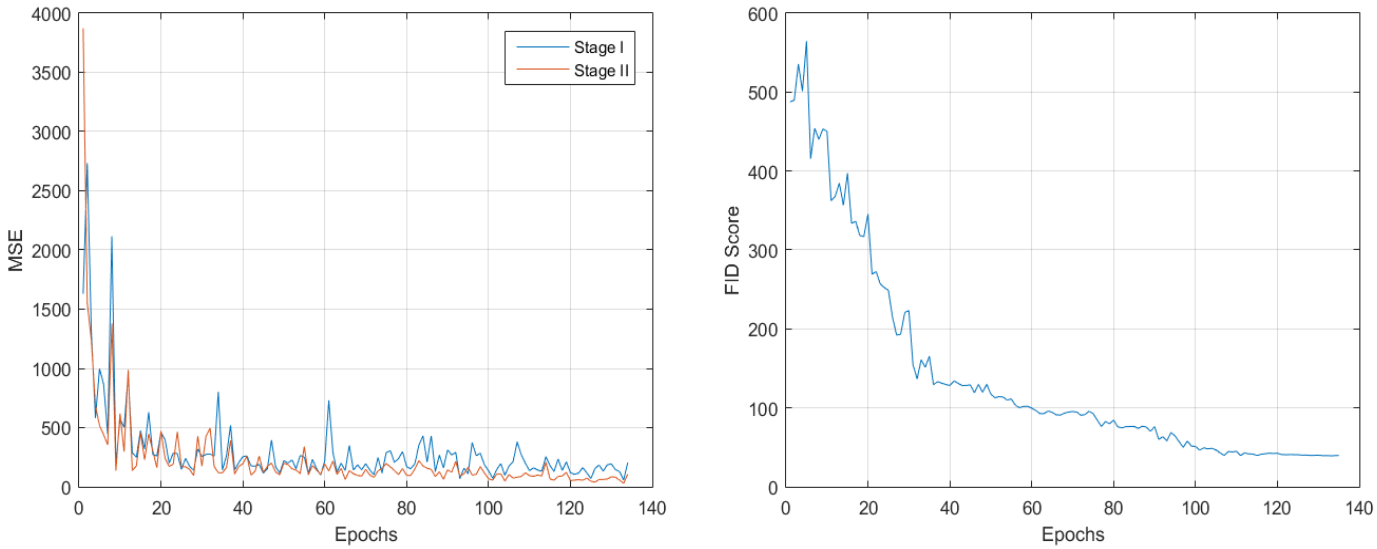


Figure 4.22 MSE vs. Epochs plot and FID vs. Epoch plot for Inter - Species Translation (Pied Kingfisher to White Throated Kingfisher)

Quantitative Analysis Graphs for Inter – Genus Species Translations

(Cattle Egret → Indian Pond Heron)

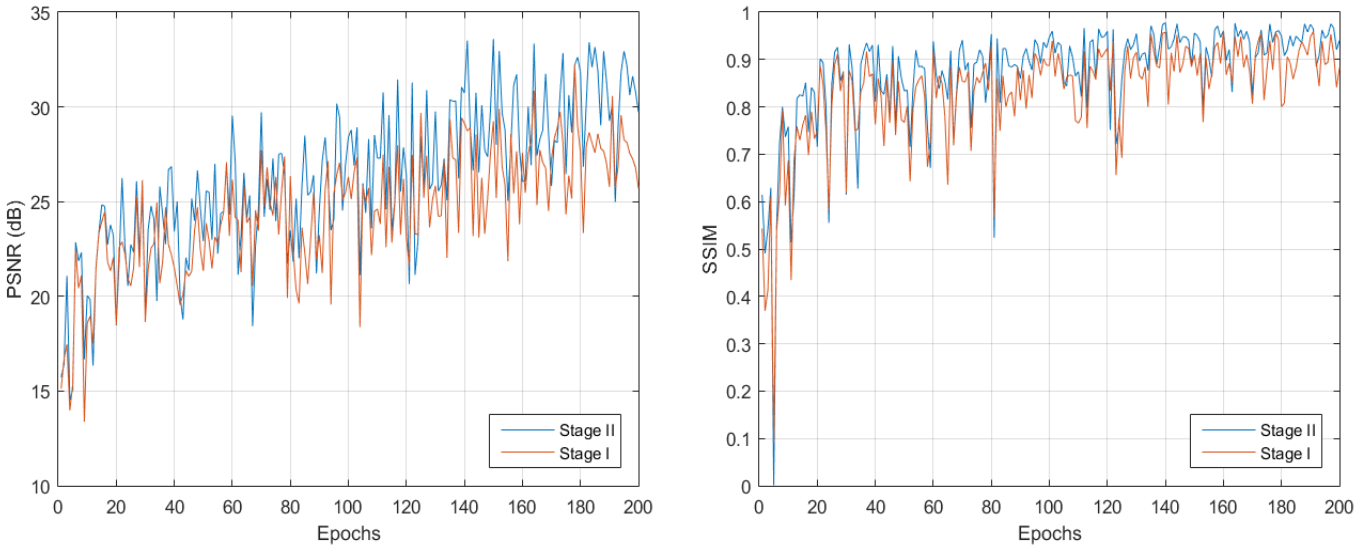


Figure 4.23 PSNR vs. Epochs plot and SSIM vs. Epochs plot for Inter - Genus Translation (Cattle Egret → Indian Pond Heron)

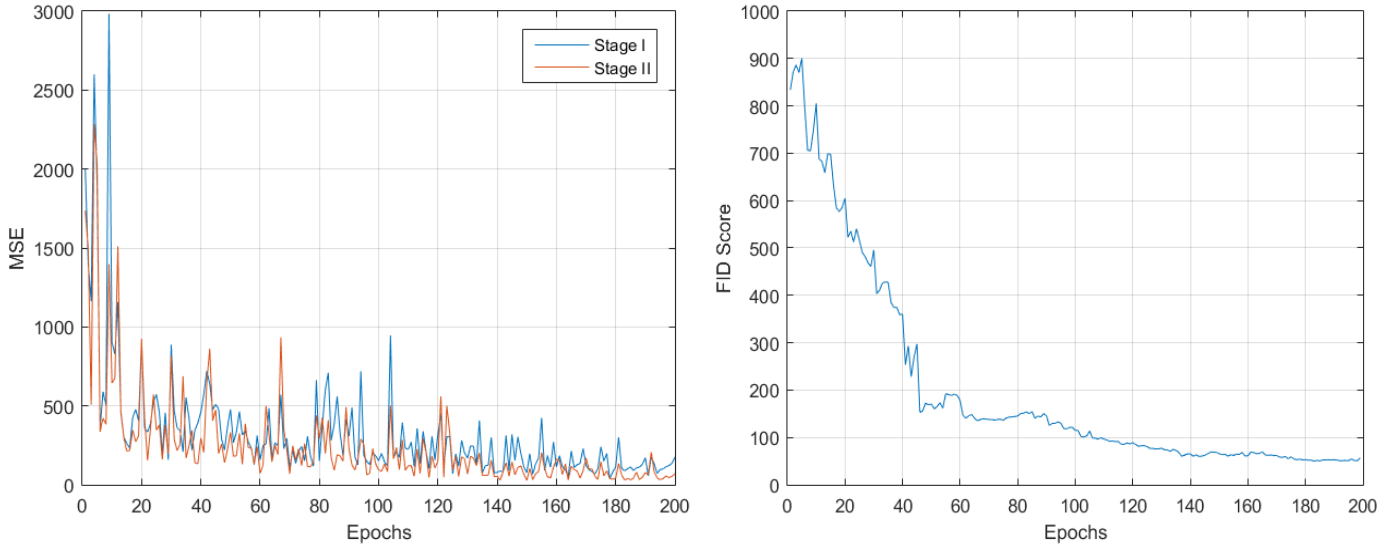


Figure 4.24 MSE vs. Epochs plot and FID vs. Epoch plot for Inter - Genus Translation (Cattle Egret → Indian Pond Heron)

Quantitative Analysis Graphs for Inter – Family Species Translations

(Oriental Magpie Robin → White Browed Wagtail)

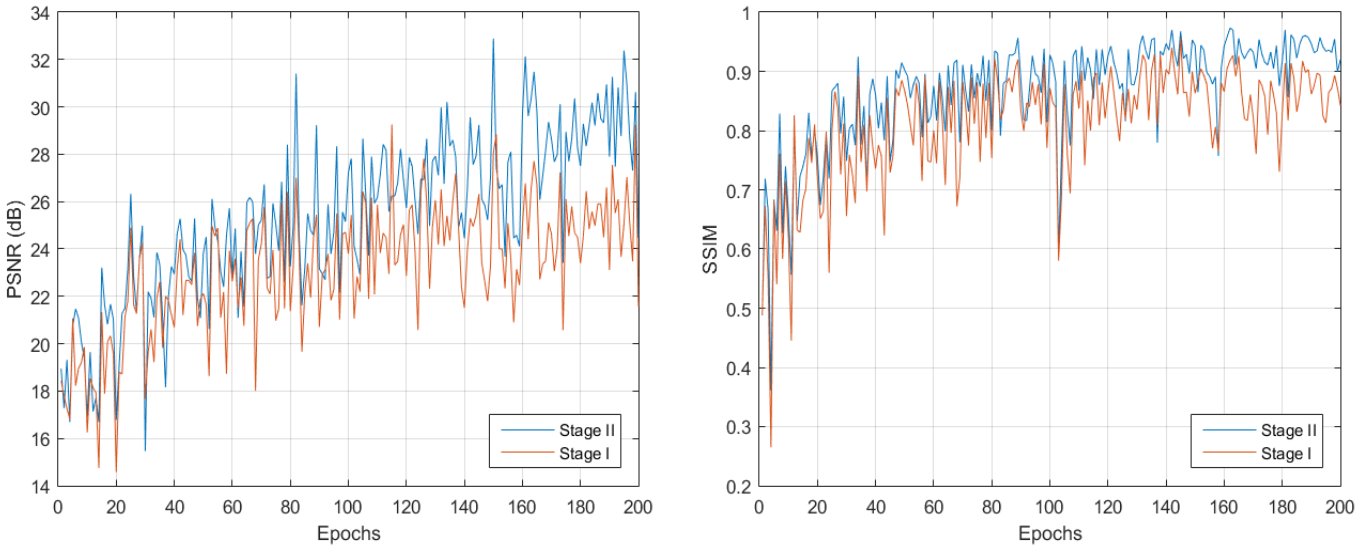


Figure 4.25 PSNR vs. Epochs plot and SSIM vs. Epochs plot for Inter - Family Translation (Oriental Magpie Robin → White Browed Wagtail)

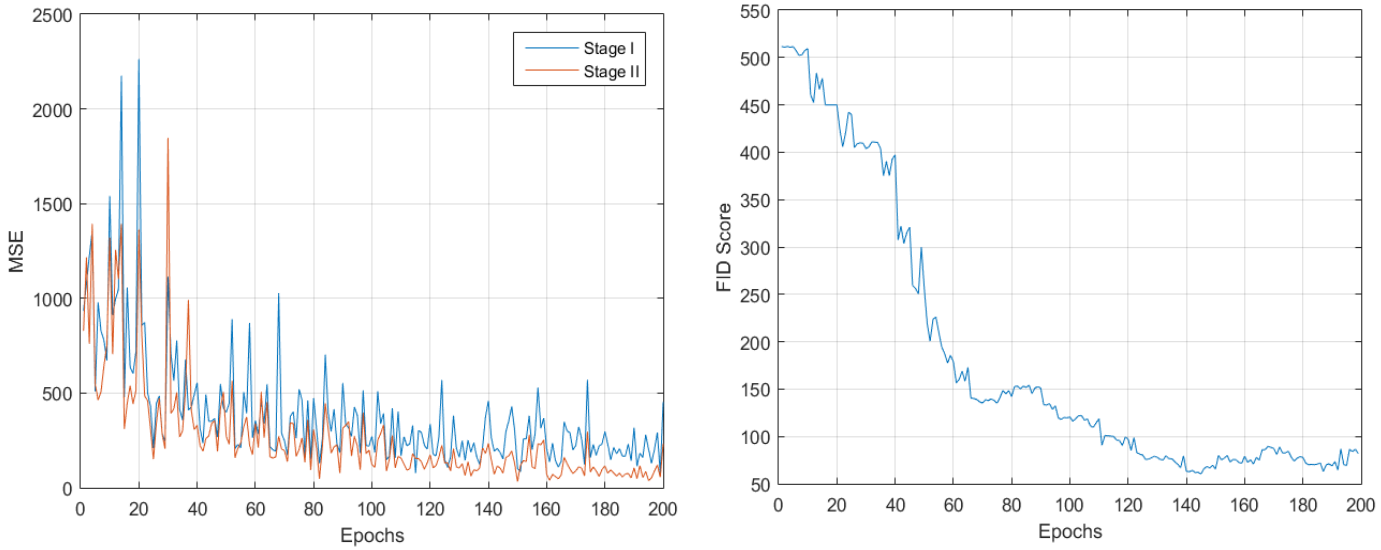
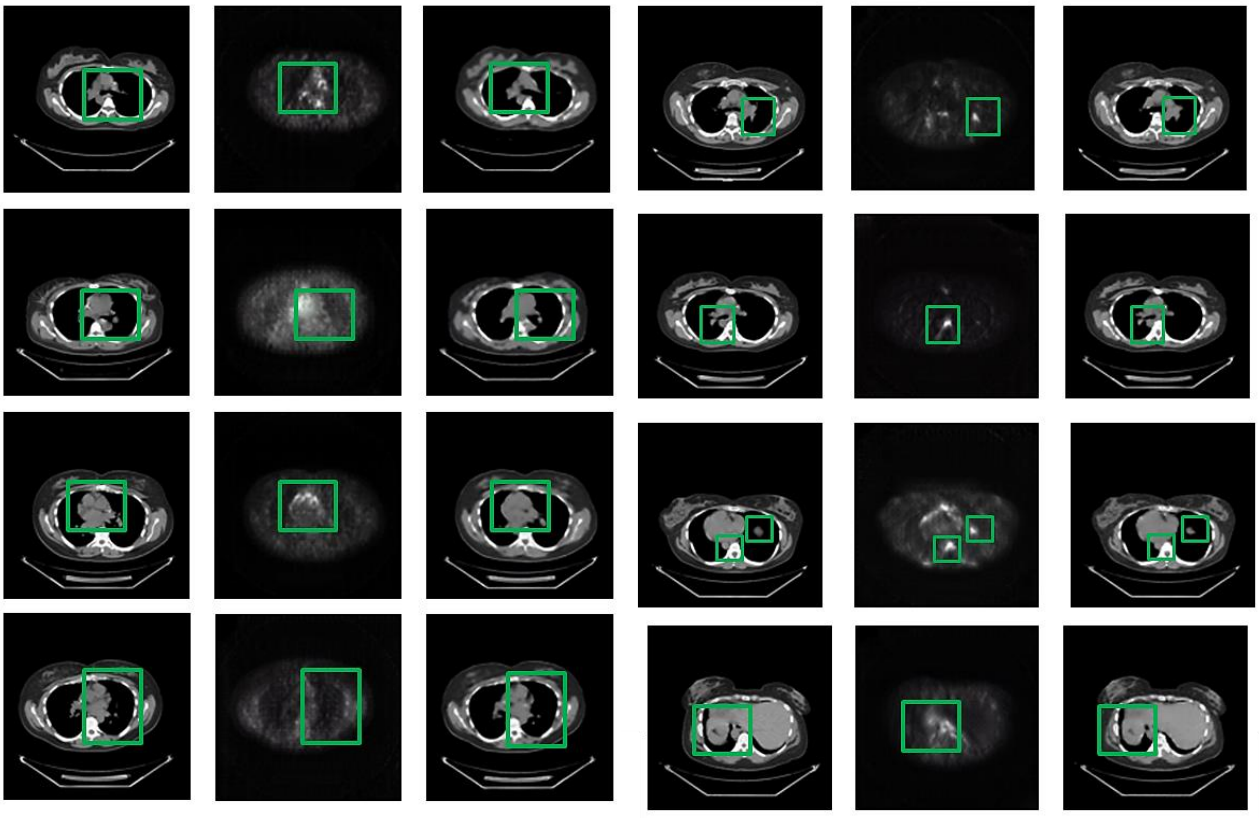


Figure 4.26 MSE vs. Epochs plot for Inter – Family Translation (Oriental Magpie Robin → White Browed Wagtail)

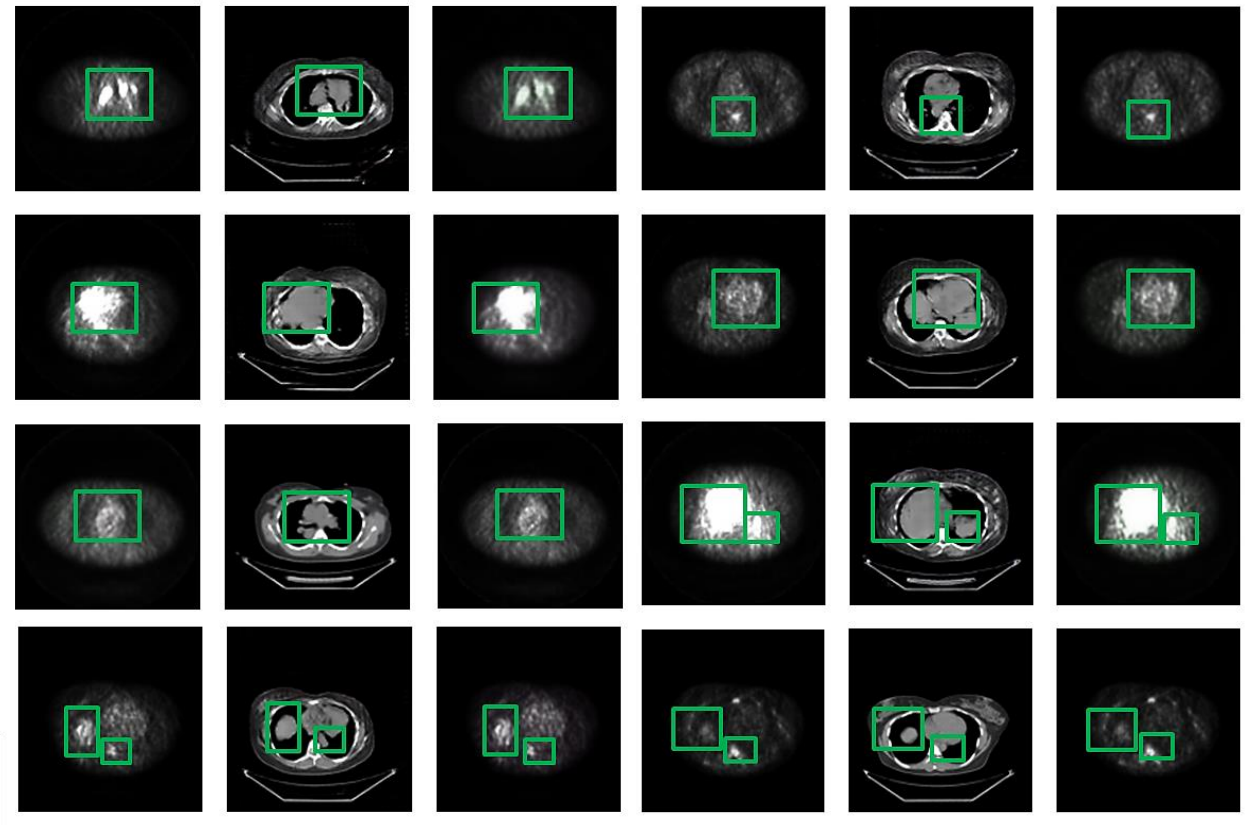
4.5.2 Simulation Results on Breast Cancer Tumour Translation

The translation results of the following study are presented in Figure 4.27 (PT → CT), Figure 4.28 (CT → PT), plots for various performance parameters in Figure 4.29 to Figure 4.31. It can be observed from Figure 4.27 and 4.28 that both translation and preservation of tumours was successful. The successful translations are highlighted in Figures for easy identification of tumours. Figure 4.29 shows the PSNR and SSIM plots for CT → PT conversion where the best PSNR and SSIM observed were 31.95 and 0.9480 respectively. Figure 4.30 shows the PSNR and SSIM plots for PT → CT conversions where the best PSNR and SSIM observed were 30.39 and 0.9322 respectively.



(a) (b) (c) (a) (b) (c)

Figure 4.27 Presented above are results obtained for CT → PT using the proposed model. The (a) column is the ground truth, (b) is the translated image and (c) is the recreated image using the translated image.



(a) (b) (c) (a) (b) (c)

Figure 4.28 Presented above are results obtained for PT → CT using the proposed model. The (a) column is the ground truth, (b) is the translated image and (c) is the recreated image using the translated image.

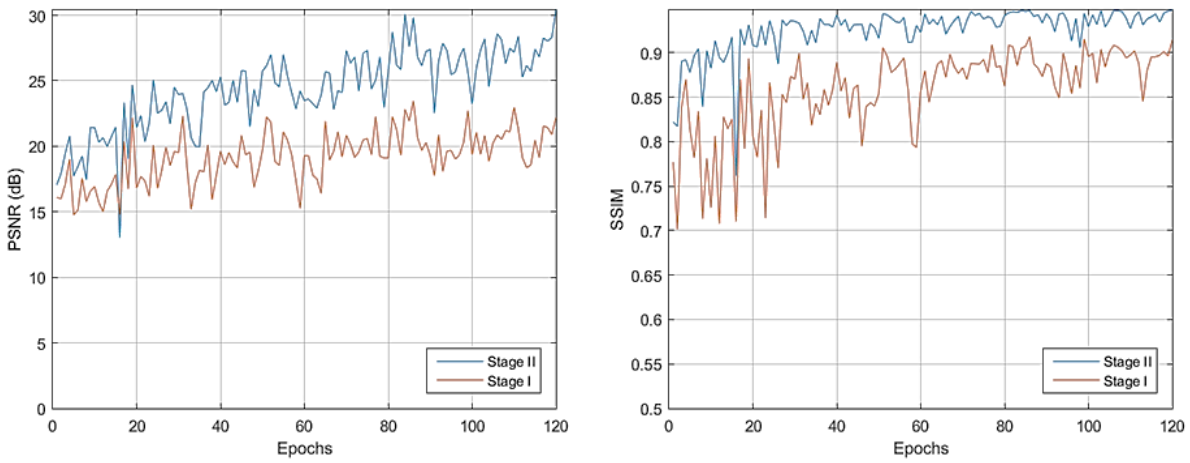


Figure 4.29 PSNR(dB) and SSIM vs Epoch Graph for PT → CT conversion.

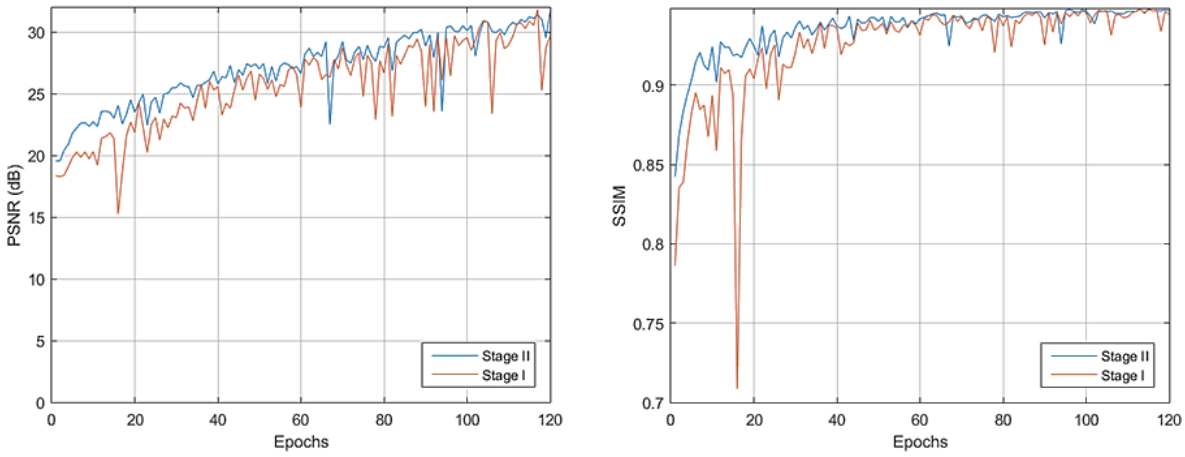


Figure 4.30 PSNR(dB) and SSIM vs Epoch Graph for CT → PT conversion.

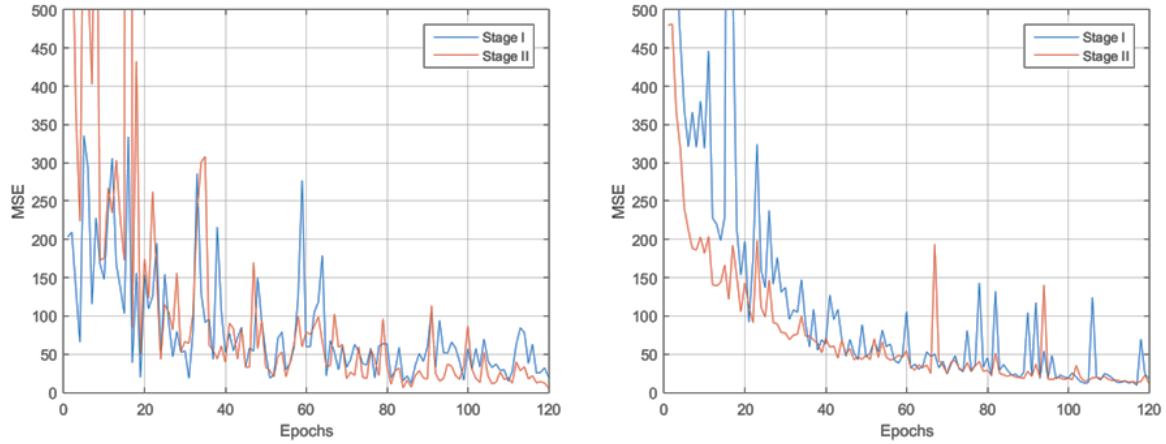


Figure 4.31 MSE vs Epoch Graph for PT → CT and CT → PT conversions.

Table 4.7 Comparison of Various Performance Parameters. The dash indicates that those parameters were not calculated in that publication by authors. The * indicates the best MSE obtained in that study.

Model	PSNR (max)	SSIM (max)	MSE (mean)	MAE (mean)	Organ
From CT → PET	–				
Ben-Cohen et al. [68]	30.4	-	-	0.72	Liver
Bi et al. [69]	28.06	-	-	4.60	Thorax
Proposed Model (Stage - I)	**30.87**	**0.9480**	**55.61**	**1.5735**	**Breast**
Proposed Model (Stage - II)	**31.95**	**0.9480**	**27.82**	**0.5961**	
From PET → CT					
Armanious et al. [72]	24.62	0.9160	264.6*	-	Brain
Proposed Model (Stage - I)	**23.45**	**0.9312**	**46.77**	**1.1126**	**Breast**
Proposed Model (Stage - II)	**30.39**	**0.9322**	**24.47**	**0.8288**	

Surprisingly, the tumours were also preserved during the recreation of input image modality. The simulation was operated for one hundred and twenty epochs (120) and took nearly twenty seven hours to complete. More number of epochs were not pursued as performance parameters like SSIM and PSNR had saturated and were oscillating between similar values. This can also be observed in the SSIM vs. epoch fig. for both the simulations. A similar trend was observed for PSNR as well. Although, MSE improved considerably, but as it had no impact on quality of images produced, further investigation was not pursued. However, slightest of misalignments in translations lead to meaningless distortions or stray artefacts in some of the images. Moreover, these artefacts may get magnified in the second stage and produce meaningless blurry images.

4.6 SUMMARY

In this chapter, a new bird dataset was introduced along with the datasets used in the study. The simulations were carried out with two models. The first model used bird images for translation and the second model is a medical image processing application developed from the first model. The application performs translations from CT scan to PET scan and vice versa, while preserving the tumours. Furthermore, the geometrical attacks applied on the images were also discussed.

The model is capable of successful translation between image modalities, preservation of tumours and resolution improvement between 2× and 4×. The results are compared with existing publications as shown in Table 4.6 and Table 4.7. It can be concluded that the model is efficient and can perform translations with many types of target images modalities; however it is sensitive to changes in various geometrical and morphological differences. The future scope and detailed conclusion of these simulations are discussed in next chapter.

CHAPTER – 5

CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION

Generative adversarial nets have become extremely popular among researchers of deep learning community due to their ability to generate images using only latent noise. Apart from image generation, GANs have been further developed to perform other deep learning tasks like image translation, resolution improvement, classification, conditional image generation, etc. Among these applications, Image translation and resolution improvement are needed in a large number of tasks and are the main focus of this dissertation.

To aid the conservation and preservation schemes for birds through deep learning, a dataset was compiled by collecting images of birds which are commonly found in north western region of the Indian subcontinent. To expand the existing bird datasets a model was proposed that used two GANs; Cycle GAN and SR GAN. The two GANs are concatenated to produce a model for style transfer (using Cycle GAN) and resolution improvement using (SR GAN). The simulation was designed to test the algorithm on birds from different species, genus and families. The task of image-to-image translation was successfully performed and better results were obtained when compared with existing models. An up-scaling factor between 2× and 4× was obtained and PSNR (33.5918), SSIM (0.9778), MSE (28.4795) and FID (38.11) were obtained which are superior to the current state of the art publications.

For the second model, medical images were used to develop an application for the task of translating breast cancer tumour CT scan and PET scan images. The proposed model was tested on PET scan and CT scan images of cancer patients using Cycle GAN for translation and SR GAN for resolution improvement. In order to preserve tumours, a feature loss and tumour loss was employed using two basic U-Net architectures. This was done to prevent tumours from vanishing after translation. When compared to the state of the art publications, higher values of PSNR (31.95), SSIM (0.9480), MSE (24.47) and MAE (0.5961) were obtained. However, the algorithm failed in cases where the appearances of the birds were very similar or if they had very different looking male and female counterparts.

Research on GAN is greatly aided by python which is one of the most favoured languages used for machine learning and deep learning tasks. It has plenty of modules for image processing and manipulation along with predefined layers and architectures of recent neural networks. Python also

utilizes the NVIDIA graphic processing units via CUDA drivers. The GPUs have dedicated CUDA cores for faster computation of algorithms.

A major drawback of GAN is its large training time. GAN trains two deep CNNs simultaneously, increasing its time complexity. To reduce time complexity, high end GPUs are required which further puts a cost constraint to usage of large models of GAN.

5.2 FUTURE SCOPE

Rapid developments in technology have lead mankind to bridge the gap between human brain and artificial intelligence while destroying balance of nature. To prevent nature from further damage, deep learning algorithms can help in conservation of biodiversity. In this age where biodiversity is vanishing at a fast pace, GAN can be used for preservation of endangered species not only in natural habitat but also in digital format for those species which are critically endangered and on verge of extinction. The future scope of this dissertation is seen as:

1. The dataset does not contain any annotations for various body parts of birds. These annotations play a vital role in image classification and recognition. Inclusion of annotations for body parts of birds will render the data set useful for species recognition as well. Also, currently the proposed data set covers only fifty common species but in near future it can be expanded to one hundred species.

2. The model can be improved by using progressively growing GAN model for resolution improvement as SR GAN requires two sets of images; LR and HR. In current simulation, it wasn't implemented as it is computationally expensive. However, if higher resolutions are desired, then progressively growing GAN might provide a higher up-scaling factor.

3. For the second model, disjoint patient image sets are yet to be tested and may provide some interesting insights.

4. Furthermore, in this simulation, only PET scan and CT scan images were used but other body parts and image modalities scan can also be incorporated, like MRI scan and mammograms.

5. The research can be further extended by using GANs for 3-D rendering and producing a full body 3D scan to better understand the out spread of cancer and dose calculation.

6. Although the results produced by GAN are extraordinary, yet the time complexity of this algorithm is too high. In future, attempts can be made to reduce the training time required by GAN.

REFERENCES

[1] Goodfellow I *et al.* (2014). Generative adversarial nets. In Advances in *neural information processing systems*, pp. 2672-2680.

[2] Crick, HQ (2004). The impact of climate change on birds. *Ibis*, 146, 48-56.

[3] La Sorte, FA and Jetz, W (2010). Avian distributions under climate change: towards improved projections. *Journal of Experimental Biology*, 213(6), 862-869.

[4] Andren, H (1994). Effects of habitat fragmentation on birds and mammals in landscapes with different proportions of suitable habitat: a review. *Oikos*, 355-366.

[5] Wolf, BO and Walsberg, GE (1996). Thermal effects of radiation and wind on a small bird and implications for microsite selection. *Ecology*, 77(7), 2228-2236.

[6] Paul, M R (2015). A review of house sparrow population decline in India. *Asia Pacific Journal of Research* Vol: I. Issue XXIX.

[7] Welinder P *et al.* (2010). Caltech-UCSD birds 200.

[8] Van Horn G *et al.* (2017). The inaturalist challenge 2017 dataset. *arXiv preprint* arXiv:1707.06642, 1(2).

[9] Berg T *et al.* (2014). Birdsnap: Large-scale fine-grained visual categorization of birds. In Proceedings of the IEEE Conference on *Computer Vision and Pattern Recognition*, pp. 2011-2018.

[10] Van Horn G *et al.* (2015). Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In Proceedings of the IEEE Conference on *Computer Vision and Pattern Recognition*, pp. 595-604.

[11] Dhillon PK *et al.* (2018). The burden of cancers and their variations across the states of India: the Global Burden of Disease Study 1990–2016. *The Lancet Oncology*, 19(10), 1289-1306.

[12] Chen SE and Williams L. (1993). View interpolation for image synthesis. In Proceedings of the 20th annual conference on *Computer graphics and interactive techniques*, pp. 279-288. ACM.

[13] Ferwerda JA *et al.* (1996). A model of visual adaptation for realistic image synthesis. In Proceedings of the 23rd annual conference on *Computer graphics and interactive techniques*, pp. 249-258. ACM.

[14] Jia J and Tang CK (2003). Image repairing: Robust image synthesis by adaptive nd tensor voting. In IEEE Computer Society Conference on *Computer Vision and Pattern Recognition*, 2003. Proceedings, Vol. 1, pp. I-I. IEEE.

[15] Zhu JY *et al.* (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the *IEEE International Conference on Computer Vision*, pp. 2223-2232.

[16] Mirza M and Osindero S (2014). Conditional generative adversarial nets. *arXiv preprint* arXiv:1411.1784.

[17] Ledig C *et al.* (2017). Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE conference on *computer vision and pattern recognition* (pp. 4681-4690).

[18] LeCun Y *et al.* (1999). Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision* (pp. 319-345). Springer, Berlin, Heidelberg.

[19] LeCun Y (1988). A theoretical framework for back-propagation. In Proceedings of the 1988 *connectionist models summer school* (Vol. 1, pp. 21-28). CMU, Pittsburgh, Pa: Morgan Kaufmann.

[20] Goodfellow, I. (2016). NIPS 2016 tutorial: Generative adversarial networks. *arXiv preprint* arXiv:1701.00160.

[21] Salimans T *et al.* (2016). Improved techniques for training gans. In Advances in *Neural Information Processing Systems* (pp. 2234-2242).

[22] Heusel M *et al.* (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Advances in *Neural Information Processing Systems*, pp. 6626-6637.

[23] Li J *et al.* (2019). A Novel Generative Model with Bounded-GAN for Reliability Classification of Gear Safety. *IEEE Transactions on Industrial Electronics*.

[24] Kim K and Myung H. (2018). Autoencoder-combined generative adversarial networks for synthetic image data generation and detection of jellyfish swarm. *IEEE Access*, 6, 54207-54214.

[25] Xuan Q *et al.* (2018). Multiview generative adversarial network and its application in pearl classification. *IEEE Transactions on Industrial Electronics*, 66(10), 8244-8252.

[26] Tan WR *et al.* (2017). ArtGAN: Artwork synthesis with conditional categorical GANs. In 2017 *IEEE International Conference on Image Processing*, pp. 3760-3764. IEEE.

[27] Tan WR *et al.* (2018). Improved artgan for conditional synthesis of natural image and artwork. *IEEE Transactions on Image Processing*, 28(1), 394-409.

[28] Reed S *et al.* (2016). Generative adversarial text to image synthesis. *arXiv preprint* arXiv:1605.05396.

[29] Tulyakov S *et al.* (2018). Mocogan: Decomposing motion and content for video generation. In Proceedings of the *IEEE conference on Computer Vision and Pattern Recognition*, pp. 1526-1535.

[30] Bhattacharjee P and Das S (2017). Temporal coherency based criteria for predicting video frames using deep multi-stage generative adversarial networks. In *Advances in Neural Information Processing Systems*, pp. 4268-4277.

[31] Yoon J, Jordon J and Van Der Schaar M (2018). Gain: Missing data imputation using generative adversarial nets. *arXiv preprint* arXiv:1806.02920.

[32] Chen H *et al.* (2018). The rise of deep learning in drug discovery. *Drug Discovery Today*, 23(6), 1241-1250.

[33] Mardani M *et al.* (2019). Deep Generative Adversarial Neural Networks for Compressive Sensing MRI. *IEEE transactions on Medical Imaging*, 38(1), 167-179.

[34] Yang G *et al.* (2018). DAGAN: deep de-aliasing generative adversarial networks for fast compressed sensing MRI reconstruction. *IEEE transactions on Medical Imaging*, 37(6), 1310-1321.

[35] Ghamisi P and Yokoya N (2018). Img2dsm: Height simulation from single imagery using conditional generative adversarial net. *IEEE Geoscience and Remote Sensing Letters*, 15(5), 794-798.

[36] Xu C *et al.* (2019). GANobfuscator: Mitigating Information Leakage under GAN via Differential Privacy. *IEEE Transactions on Information Forensics and Security.*

[37] He Y *et al.* (2019). Multi-task gans for view-specific feature learning in gait recognition. *IEEE Transactions on Information Forensics and Security*, 14(1), 102-113.

[38] Pang Y, Xie J and Li X. (2018). Visual Haze Removal by a Unified Generative Adversarial Network. *IEEE Transactions on Circuits and Systems for Video Technology.*

[39] Saito Y, Takamichi S and Saruwatari H (2018). Statistical parametric speech synthesis incorporating generative adversarial networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(1), 84-96.

[40] Denton E L, Chintala S and Fergus R (2015). Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems*, pp. 1486-1494.

[41] Radford A, Metz L and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint* arXiv:1511.06434.

[42] Donahue J, Krähenbühl P and Darrell, T. (2016). Adversarial feature learning. *arXiv preprint* arXiv:1605.09782.

[43] Chen X *et al.* (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2172-2180.

[44] Zhao J, Mathieu M and LeCun Y (2016). Energy-based generative adversarial network. *arXiv preprint* arXiv:1609.03126.

[45] Mao X *et al.* (2017). Least squares generative adversarial networks. In Proceedings of the *IEEE International Conference on Computer Vision*, pp. 2794-2802.

[46] Arjovsky M, Chintala, S and Bottou L. (2017). Wasserstein gan. *arXiv preprint* arXiv:1701.07875.

[47] Berthelot D, Schumm T and Metz L (2017). Began: Boundary equilibrium generative adversarial networks. *arXiv preprint* arXiv:1703.10717.

[48] Gulrajani I *et al.* (2017). Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pp. 5767-5777.

[49] Mao X *et al.* (2018) On the Effectiveness of Least Squares Generative Adversarial Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. doi: 10.1109/TPAMI.2018.287204.

[50] Kodali N *et al.* (2017). On convergence and stability of gans. *arXiv preprint* arXiv:1705.07215.

[51] Miyato T *et al.* (2018). Spectral normalization for generative adversarial networks. *arXiv preprint* arXiv:1802.05957.

[52] Zhang H *et al.* (2018). Self-attention generative adversarial networks. *arXiv preprint* arXiv:1805.08318.

[53] Jolicoeur-Martineau A (2018). The relativistic discriminator: a key element missing from standard GAN. *arXiv preprint* arXiv:1807.00734.

[54] Zhang H *et al.* (2017). Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In Proceedings of the *IEEE International Conference on Computer Vision*, pp. 5907-5915.

[55] Zhang H *et al.* (2017). Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *arXiv preprint* arXiv:1710.10916.

[56] Isola P *et al.* (2017). Image-to-image translation with conditional adversarial networks. In Proceedings of the *IEEE conference on Computer Vision and Pattern Recognition*, pp. 1125-1134.

[57] Yi Z *et al.* (2017). Dualgan: Unsupervised dual learning for image-to-image translation. In Proceedings of the *IEEE International Conference on Computer Vision* (pp. 2849-2857).

[58] Kim T *et al.* (2017). Learning to discover cross-domain relations with generative adversarial networks. In Proceedings of the *34th International Conference on Machine Learning*-Vol 70, pp. 1857-1865.

[59] Isola P *et al.* (2017). Image-to-image translation with conditional adversarial networks. In Proceedings of the *IEEE conference on Computer Vision and Pattern Recognition*, pp. 1125-1134.

[60] Ledig C *et al.* (2017). Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the *IEEE conference on Computer Vision and Pattern Recognition*, pp. 4681-4690.

[61] Karras T *et al.* (2017). Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint* arXiv:1710.10196.

[62] Wang X *et al.* (2018). ESR GAN: Enhanced super-resolution generative adversarial networks. In *European Conference on Computer Vision*, pp. 63-79. Springer, Cham.

[63] Nie D *et al.* (2017). Medical image synthesis with context-aware generative adversarial networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 417-425. Springer, Cham.

[64] Wolterink JM *et al.* (2017). Deep MR to CT synthesis using unpaired data. In *International Workshop on Simulation and Synthesis in Medical Imaging*, pp. 14-23. Springer, Cham.

[65] Zhao M *et al.* (2018). Craniomaxillofacial Bony Structures Segmentation from MRI with Deep-Supervision Adversarial Learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 720-727. Springer, Cham.

[66] Chartsias A *et al.* (2017). Adversarial image synthesis for unpaired multi-modal cardiac data. In *International Workshop on Simulation and Synthesis in Medical Imaging*, pp. 3-13. Springer, Cham.

[67] Jiang J *et al.* (2018). Tumour-aware, adversarial domain adaptation from ct to mri for lung cancer segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 777-785. Springer, Cham.

[68] Ben-Cohen A *et al.* (2019). Cross-modality synthesis from CT to PET using FCN and GAN networks for improved automated lesion detection. *Engineering Applications of Artificial Intelligence*, 78, 186-194.

[69] Bi L *et al.* (2017). Synthesis of positron emission tomography (PET) images via multi-channel generative adversarial networks (GANs). In *Molecular Imaging, Reconstruction and Analysis of Moving Body Organs, and Stroke Imaging and Treatment*, pp. 43-51. Springer, Cham.

[70] Jin CB *et al.* (2019). Deep CT to MR synthesis using paired and unpaired data. *Sensors*, 19(10), 2361.

[71] Emami H *et al.* (2018). Generating synthetic CTs from magnetic resonance images using generative adversarial networks. *Medical Physics*, 45(8), 3627-3636.

[72] Armanious K *et al.* (2018). MedGAN: medical image translation using GANs. *arXiv preprint* arXiv:1806.06397.

[73] Wei W *et al.* (2018). Learning myelin content in multiple sclerosis from multimodal MRI through adversarial training. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 514-522. Springer, Cham.

[74] Choi H and Lee DS (2018). Generation of structural MR images from amyloid PET: Application to MR-less quantification. *Journal of Nuclear Medicine*, 59(7), 1111-1117.

[75] Gonzalez RC and Woods RE (1992). Digital Image Processing Addison-Wesley. Reading, Ma, 2.

[76] Nair V and Hinton GE (2010). Rectified linear units improve restricted boltzmann machines. In Proceedings of the *27th international Conference on Machine Learning*, pp. 807-814.

[77] Xu B *et al.* (2015). Empirical evaluation of rectified activations in convolutional network. *arXiv preprint* arXiv:1505.00853.

[78] Scherer D, Müller A and Behnke S (2010). Evaluation of pooling operations in convolutional architectures for object recognition. In *International Conference on Artificial Neural Networks*, pp. 92-101. Springer, Berlin, Heidelberg.

[79] Ioffe S and Szegedy C (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint* arXiv:1502.03167.

[80] Ulyanov D, Vedaldi A and Lempitsky V (2016). Instance normalization: The missing ingredient for fast stylization. *arXiv preprint* arXiv:1607.08022.

[81] Xu L *et al.* (2014). Deep convolutional neural network for image deconvolution. In *Advances in Neural Information Processing Systems*, pp. 1790-1798.

[82] He K *et al.* (2016). Deep residual learning for image recognition. In Proceedings of the *IEEE conference on Computer Vision and Pattern Recognition*, pp. 770-778.

[83] Ronneberger O, Fischer P and Brox T (2015). U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234-241. Springer, Cham.

[84] Wang Z *et al.* (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on Image Processing*, 13(4), 600-612.

[85] Hore A and Ziou D (2010). Image quality metrics: PSNR vs. SSIM. In 2010 *20th International Conference on Pattern Recognition*, pp. 2366-2369. IEEE

[86] Clark K *et al.* (2013) The Cancer Imaging Archive (TCIA): Maintaining and Operating a Public Information Repository, Journal of Digital Imaging, Vol. 26, No. 6, pp 1045-1057.

[87] Kostakoglu L *et al.* (2015) ACRIN 668 Investigative Team. A Phase II Study of 3'-Deoxy-3'-18F-Fluorothymidine PET in the Assessment of Early Response of Breast Cancer to Neoadjuvant Chemotherapy: Results from ACRIN 6688. Journal of Nuclear Medicine, 56(11):1681-9. doi: 10.2967/jnumed.115.160663.

[88] Agustsson E and Timofte R (2017). Ntire 2017 challenge on single image super-resolution: Dataset and study. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 126-135.

APPENDIX

A Species Wise List of Bird Images in CBNWI – 50

S. No.	Common Name	Original Images	Images with attacks	Total
1.	Asian Koel	66	396	462
2.	Bank Myna	50	300	350
3.	Black Drongo	83	498	581
4.	Black Kite	121	726	847
5.	Black Rumped Flame Back Woodpecker	50	300	350
6.	Black Winged Stilt	165	990	1155
7.	Brahminy Myna	107	642	749
8.	Cattle Egret	212	1272	1484
9.	Common Myna	252	1512	1764
10	Common Crow	108	648	756
11.	Common Moorhen	50	300	350
12.	Common Sandpiper	50	300	350
13.	Common Tailor Bird	61	366	427
14.	Common Teal	63	378	441
15.	Coopersmith Barbet	50	300	350
16.	Eurasian Coot	51	306	357
17.	Eurasian Collared Dove	169	1014	1183
18.	Great White Pelican	50	300	350
19.	Greater Coucal	46	276	322
20.	Green Bee Eater	134	804	938
21.	Hoopoe	50	300	350
22.	Indian Grey Hornbill	71	426	497

23.	Indian House Sparrow	100	600	700
24.	Indian Pond Heron	176	1056	1232
25.	Indian Robin	100	600	700
26.	Indian Roller	50	300	350
27.	Jungle Babbler	147	882	1029
28.	Large Grey Babbler	100	600	700
29.	Laughing Dove	255	1530	1785
30.	Lesser Cormorant	66	396	462
31.	Northern Shoveler	50	300	350
32.	Oriental Magpie Robin	182	1092	1274
33.	Peacock	250	1500	1750
34.	Pied Kingfisher	59	354	413
35.	Pied Myna	50	300	350
36.	Plum Headed Parakeet	50	300	350
37.	Purple Moorhen	50	300	350
38.	Purple Sunbird	178	1068	1246
39.	Red Vented Bulbul	123	738	861
40.	Red Wattled Lapwing	100	600	700
41.	Rock Pigeon	150	900	1050
42.	Roofus Treepie	77	462	539
43.	Rose Ring Parakeet	193	1158	1351
44.	Rosy Starling	98	588	686
45.	Shikra	74	444	518
46.	Spot Billed Duck	50	300	350
47.	Spotted Owlet	50	300	350

48.	White Browed Wagtail	50	300	350
49.	White Throated Kingfisher	165	990	1155
50.	Yellow Footed Green Pigeon	50	300	350
	TOTAL	5,102	30,612	35,714

LIST OF PUBLICATIONS

1. Sharma, A. and Jindal, N. "CBNWI-50: A Deep Learning Bird Dataset for Image Translation and Resolution Improvement using Generative Adversarial Network". In proceedings of *Innovations in Communication Computing and Sciences*, CGC Landran, Chandhigarh, India (July 27th – 28th, 2019), Scopus Indexed.
2. Sharma,A., Jindal, N. and Thakur, A. "Comparison on Generative Adversarial Networks – A Study". In proceedings of *International Conference on Secure Cyber Computing and Communications*, NIT Jalandhar, India (December 16th – 18th, 2019), IEEE.
3. Sharma, A. and Jindal, N. "Cross-Modality Breast Image Translation with Improved Resolution using Generative Adversarial Networks", *Wireless Personal Communication*, Springer. Communicated. (SCI Indexed)
4. Sharma, A., Jindal, N. and Rana, P. S., "Advances in Generative Adversarial Net Algorithms for Image and Video Processing", *Multimedia Tools and Applications*, Springer. Communicated. (SCIE Indexed)

ME Thesis

ORIGINALITY REPORT

8%	2%	5%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Asha Anoosheh, Eirikur Agustsson, Radu Timofte, Luc Van Gool. "ComboGAN: Unrestrained Scalability for Image Domain Translation", 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2018 Publication	<1%
2	Submitted to Imperial College of Science, Technology and Medicine Student Paper	<1%
3	"Computer Vision – ECCV 2018 Workshops", Springer Science and Business Media LLC, 2019 Publication	<1%
4	export.arxiv.org Internet Source	<1%
5	COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering, Volume 31, Issue 6 (2012-11-10) Publication	<1%

Akanksha Neeru 15/7/19