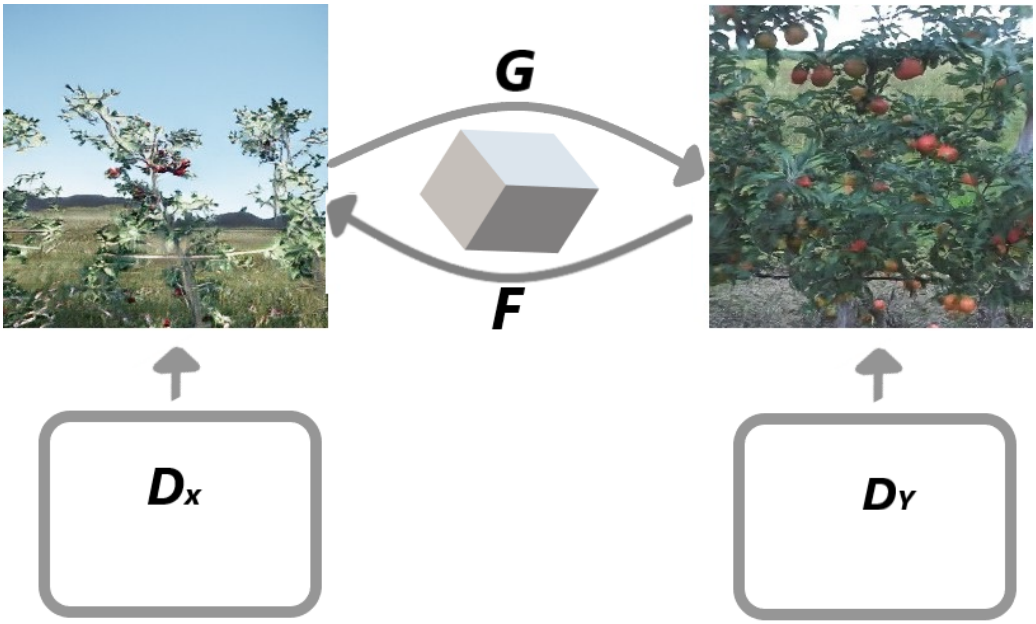
Robotics Research Lab
Department of Computer Science
Technische Universität Kaiserslautern

Master Thesis



Style Transfer Using GANs

Zulfiqar Ibrahim

February 28, 2022

Master Thesis

Style Transfer Using GANs

Robotics Research Lab
Department of Computer Science
Technische Universität Kaiserslautern

Zulfiqar Ibrahim

Day of issue : 31.08.2021
Day of release : 01.03.2022

First Reviewer : Prof. Dr. Karsten Berns
Supervisor : M.Sc. Tim Dallman

Hereby I declare that I have self-dependently composed the Master Thesis at hand. The sources and additives used have been marked in the text and are exhaustively given in the bibliography.

February 28, 2022 – Kaiserslautern

(Zulfiqar Ibrahim)

Acknowledgement

First of all, Praise to almighty ALLAH for providing me strength and endurance to complete my studies.
I like to pay my respect and gratitude to Prof. Dr. Karsten Berns for allowing me to write my thesis at Robotics Research Lab. I would also like to thank my supervisor Tim Dallman for his constant unfathomable support throughout my thesis. He has provided an environment where I can explore my topic with freedom and guided me towards the conclusion.
I would also like to pay my regards to associates at RR Lab for providing me technical assistance in these times.
In these testing times, I must pay special thanks to my Wife Malika Rafiq who supported me immensely during my stay at University.
Lastly, I am indebted to my parents and my siblings for providing me with an opportunity to study at TU Kaiserslautern. I would always be thankful to them for this generous cooperation and support.

Abstract

Precision farming has observed tremendous impacts in the field of agriculture. Farmers are trying to exploit these latest A.I and robotic based technologies for increase in the crop yield. Many companies are competing each other for autonomous vehicular robots which can automate many human-based tasks in the farming. An autonomous robotics systems need imaging data in order to train their models for various tasks like classification, segmentation and tracking. It is challenging to collect imaging data related to the crop life cycle. This work explores the use of Generative Adversarial Networks (GANs) for the generation of complex images related to the apple orchard field. The deep generative model must transfer the spatial information from training images to generated images. In the training images, there are two sets of images where one set is designed in unreal engine software and another set contains real images of the apple orchard field. A GAN is a combination of two neural networks which work in competition with each other for the construction of the images where one network's gains are the other network's losses. GANs are prone to instability in terms of training and evaluation due to their complexity and unsupervised learning nature. To subdue this instability, deep cyclic GANs are deployed to generate data from training images instead of from latent noise. This system automatically discovers patterns in the training data and emulates them for the generation of new data. This thesis explores the idea of using GANs for the generation of apple images from the apple orchard. These are very complex images with a varied distribution of data amongst them. For the first set of training data collection, two cameras are connected to an autonomous robot at two different angles that take images of apple trees. For the second set of training data , an unreal engine is used for the creation of farm like simulated environment in which 3D models of apple trees are placed to produce 3D images. In the generation of 3D data through the unreal engine, there is a leverage on the choice of any settings like lighting, weather, apple size, etc. This allows to map any data from any domain to the apple tree domain and also transfer the style from synthetic data to target data. The network is trained on a dataset which is the combination of synthetic data and real data. The view angle of synthetic images must be set similar to the original image's view angle. This allows both datasets to be semantically similar. Several experiments have been performed with varied environments in synthetic data and an inference score is calculated for the evaluation of the final results. The trained model of cyclic GANs can generate countless images apple orchard which can be used as training data for other agriculture-based machine learning tasks.

Contents

1. Introduction

After the recent surge in information technology and artificial intelligence, farmers around the globe have begun deploying such technologies to increase the maximum yield of their crops. A.I is defined as the ability of the computer system to act like a human to provide the solution of the problem through observing and learning. Machine learning is considered as a subset of A.I in which, mathematical models are utilized to extract the patterns from the underlying data. These patterns guide the computers for better learning, without the influence of direct instructions. Generative adversarial networks (GANs) are a subset of machine learning in which they are used to generate the data from noise. The need for GANs aroused when researchers believed the shortages of the data hampering the performances of machine learning-based tasks. In the field of agriculture, it is difficult to acquire imaging data due to economic costs, weather constraints, etc. GANs can help in this area to generate crop-based imaging data for the machine learning tasks. These tasks also help the autonomous robot in the crop field for better situational awareness. The idea of this thesis is to design 3D Apple orchard field simulated in an unreal engine. The art design can be transferred from 3D images to real images which allows the creation of synthetic data in any setting.

Deep learning paradigms are used in Artificial Intelligence(A.I) to extract abundant hierarchical patterns that contain probability distributions over types of data present in intelligence-based applications [LeCun 15]. These probability distributions can be categorized as natural images, symbols in natural language corpora, or audio wave-forms which contain human speeches. In the early days of machine learning research and development, deep learning was mainly utilized for classification problems but as the technologies progressed it was observed that such networks can also be used to generate the data. The deep networks are always in need of varied data and collection of the data is very labor-intensive and expensive. Therefore to overcome this problem, researchers invented GANs which are designed to model the Real data distribution. Discriminative Models have also seen success in deep learning due to their ability to map high dimensional inputs to class labels. This happens primarily due to backpropagation and dropout algorithms through piece-wise linear units which act as obedient gradients. On the contrary, generative models with deep networks had less impact, due to difficulty in approximating the probabilistic computation presented in maximum likelihood estimation. To overcome these issues,

researchers have purposed a new method in which; a generative model sets against an adversary model. An adversary model is a discriminative model that learns to discover whether the data belongs to model distribution or the real data distribution [Cai 21].

In recent years, GANs have witnessed an increase in popularity among the AI community. At the abstraction level, GANs are neural networks that learn to re-generate the data from the given dataset. Two neural networks are set against each other where one network (generator) learns to generate data instance and another network (discriminator) learns to distinguish between true data and output data from the generator. GANs work with two networks because it is infeasible to design a function that informs about the likelihood of generated data instances from training distribution. To achieve this task, a separate neural network is required for the assessment of the generated data instances. Both networks are differentiable therefore a gradient is used to steer networks in the right direction. Given a dataset that consists of data instances X and label Y then the Generative model learns to capture joint probability $P(X|Y)$ while Discriminative models capture conditional probability $P(Y|X)$. In this setting, the generator is an important part that must learn to produce data from real latent sample space which is indistinguishable from the training distribution. The discriminator is simply an adaptive loss function which after the training completes, gets discarded. Scientists have used GANs for multiple purposes like data generation, translation of an image from one domain to another domain, image super-resolution, text-to-image, speech synthesis, and many more. These use cases have allowed researchers to further automate the tasks in the digital world. [Goodfellow 14]

Image-to-Image translation belongs to the category of computer vision and graphics problems where the main aim is to learn the mapping between input and output images by using training data, consisting of aligned image pairs. This image-to-image translation allows to convert one image to another image in a more comprehensive way but due to the unavailability of the paired data most of the time, it is difficult to utilize the full potential of such algorithms. To resolve this problem, the researchers presented an approach to learn to render the image from one domain to another domain in the absence of paired dataset. For example, if there is a picture of the village in the summer season with river and green grass and blue sky. Now consider the same image of the village in the winter season. There are noticeable stylist differences in these two images where the grass is less green and the color of the sky is different. There are clear stylist differences between these two images. A system can learn these differences and collect special characteristics from training image collection and translate them from one image to another image without the need for paired examples. This problem is known as image-to-image translation in which one image is converted from one representation of a scene, x, to a different scene y, e,g., grayscale image to color image or image to semantic labels.

Scientists [Isola 17] have provided powerful image-based translation systems in a supervised environment after years of research and development. These translation systems are the product of research in computational photography, computer vision, image processing, and computer graphics. These image-based translations required paired dataset, which is unfeasible to acquire and expensive, for example, there are only a couple of small datasets available which can be utilized for semantic segmentation. Graphics tasks like artistic stylization required input-output paired images which is more difficult since desired output requires the supervision of an artist. By diving into this research, it can be assumed that translation between one domain to another domain is possible without paired dataset.

There is a fundamental relationship between the domains. To further explore this concept, it is possible to exploit supervision at the level of sets where one set belongs to domain X and the other set belongs to Y. A general mapping $G : X \rightarrow Y$ can be learned such that $\hat{y} = G(x)$, where $x \in X$. The adversarial network is trained to differentiate between $\hat{y}$ and $y \in Y$. The mapping G must be stochastic since ideally output result of $\hat{y}$ belongs to empirical distribution $p_{data}(y)$. Therefore optimal G translates the X domain to $\hat{Y}$ domain however this learned mapping does not guarantee the individual mapping between one input instance of x to individual y in a comprehensive way. There are many mapping G that can induce the same probability distribution over $\hat{y}$. Researchers have learned that in practice, it is infeasible to optimize the adversarial network separately which leads to the problem like mode collapse.

The mode collapse is very common in the training of the GANs and is defined as when all the input x map to the same y without any learning. To resolve this problem, [Srivastava 17] researchers have developed many techniques, for example, Implicit Maximum Likelihood Estimation (IMLE) in which distance between the generated data point and real data point is calculated and move the model towards the closest generated data point. We have hyper-parameter tuning, which allows GANs to train without mode collapse. However, this technique requires patience. Cyclic consistency is another famous technique in which generated data from the first domain to the Second domain is converted back to the first domain to check the differences, which allows the GANs to be trained under the extra supervision of "Cyclic Consistency"

Deep generative networks can be used for the generation of imaging data distribution related to agriculture, which can help achieve better performances from machine learning-related tasks. Precision farming technology has gained immense popularity among the farmers of the world. In this technology, a farmer can utilize technological instruments to extract vital data from his/her crop field. These instruments are camera autonomous attached-drones, geographical information system (GIS), yield monitors, yield maps, and remote sensors. When autonomous vehicular systems are deployed for farming tasks, then they require a huge amount of data to perform machine learning-related functions. The availability of agricultural-based data for autonomous systems is a challenging task to achieve. This data can belong to imagine category or to the numerical category. A GAN creates the synthetic imaging data coming from a similar probability distribution of agricultural-based data. Autonomous robotic systems can use synthetic data for classification, segmentation, and detection-related tasks. Therefore, this thesis will investigate different state-of-the-art GANs to achieve high-quality images of the Apple orchard field. It is important to know the growth phase of crop is time series dependent, which means the color, shape and size of crops changes in equal interval of time. A GAN can capture these changes and replicate the style in new imaging data.

1.1 Motivation

Food is the primary source of nourishment for humans and animals, and the production of this sustenance is considered a cerebral pillar for the world's economy. The majority of the countries consider Agriculture to be their primary source of employment. The advancements in A.I and robotics have opened a door for the automation of agriculture-related tasks which can increase efficiency and crops yield. Many countries rely on

traditional methods of farming because farmers are hesitant to adopt these technologies due to lack of knowledge, exuberant costs, and unavailability of technical services. Generally, there are three categories in the process of farming which are pre-harvesting, harvesting, and post harvesting. Machine learning can play a crucial role in all these three stages of agriculture and reduce the losses by providing rich recommendations and stats about the crop. Many technological-based companies have ventured into agriculture. However, there is a long road ahead for the full impact of such technologies. Nearly 40% of crop's yield discarded due to diseases and pest infestation in the world. There is a dire need for the deployment of these technologies for the protection of crops from diseases and contribute to the eradication of world hunger [Meshram 21].

Generative Adversarial Networks (GANs) can play important role in the generation of agriculture-based data. This synthetic data can be used for training and evaluation purposes and can have an effective impact on the performances of the networks. One of the use cases of robotics and machine learning is to utilize an autonomous robot vehicle that can traverse in the field and attached high def industrial-scale camera is taking live feed and pre-processing the images using techniques like image processing, computer vision, and the internet of things (IoT) then trained models are used for any machine learning tasks. These tasks help the farmers to get better statistics of farms e.g detection of disease, area of crop infection, the growth stage of the crop, the overall health of the crop, etc. A good crop-based imaging dataset is very hard to find. Many companies are contributing to increasing the quality datasets for agriculture-based applications.

There are many challenges in the creation of a dataset based on farming. Every soil has different compositions like pH level, soil fertility, the contribution of Phosphorus(P), Potassium (K), Moisture content (MC), and Organic Carbon (OC). All these features and different quality of seeds lead to crops with different features like color, size, tastes. It is infeasible to use an international crop dataset for machine learning tasks for local applications [Liakos 18]. Therefore researchers must find a way to generate a local dataset from local farms which can perform better compared to globally available datasets. For the creation of a synthetic dataset, we have two approaches. The first approach is to use 3D creation tools like blender where handcrafted 3D models are designed for the specific crop and place these models in a simulated environment like in unreal engine [Sanders 16]. This simulation allows changing the weather, day cycle, or other environment feature to get diverse imagine data. This approach allows researchers to collect the data in any weather condition without specifically visiting the farm for data collection. In this technique, a professional 3D Designer can handcraft the features for crop model however this process is substantially expensive since professional designers are not easy to come by. In the second approach, GANs can be used for the creation of a dataset however there is still the requirement of 3d artists who can create minimum 3D models. These 3D models are used in simulation from Unreal Engine for the collection of data. Researchers can create farm-like settings, place this corps model in it and a camera attached to a robotic 3D vehicle can take images according to requirement. Researchers also need real image data from the farm for the mapping process. GANs can now learn the mapping between synthetic data and real data after the training process and style can also be transferred from synthetic to generated images. Designers can create any kind of features about crops and adjust the learned mapping between synthetic and real data. These trained GANs are used to generate data in abundance, as per the requirement of the given machine learning

tasks.

1.2 Tasks

In this thesis, Cyclic GANs have been deployed to re-create the imaging data from the apple orchard field. The main task of cyclic GANs is to translate the image from two domains without paired data for training. The training dataset consists of two sets of images where one dataset is considered as synthetic data and the second dataset is categorized as real data. For synthetic data, 3D models of apple trees including apples are created which are then placed in Unreal Engine's farm-like environment. A built-in camera is then utilized to capture the images of apple trees under user-controlled sunlight. For real data, a real apple orchard field is visited for the collection of images. A remote-controlled robot vehicle is used where two ZED cameras are mounted and images are captured at a different angle. The video feed from these cameras is recorded at 60 Fps which is then used to extract the imaging data. ZED Cameras also allow saving the depth information from video feed which plays a very important role for the selection of region..

There is a requirement for a special imagine pipeline that extracts the images from the depth map of ZED cameras. A depth map from ZED Cameras allows the selection of the region of interests (ROI) that are closer to the camera. For the dataset, only those regions are important which contain maximum information of apple trees. Many image processing/OpenCV techniques are available which can be used to crop an area within an image where maximum depth in formations exits. Once the ROI is cropped, it can be saved in a real dataset.

In the aforementioned sections, the significance of Cyclic GANs will be discussed for the image-to-image translation without paired data. All the mathematical techniques will also be thoroughly dissected to gain a better understanding. The training process will be deeply observed for Cyclic GANs for the given dataset.
Basic Overview of the tasks are:

- Creation of ground-truth dataset comprises of apple trees images, with the help of OpenCV-based imagine pipeline.
- Creation of synthetic dataset comprises of apple trees images from Unreal Engine's simulation.
- Implementation of Cyclic GANs architecture which consists of Generator and Discriminator.
- Evaluation of generated results from implemented model using Fréchet inception distance (FID) metric.

2. Background

This chapter deals with the theoretical background of the deep generative network which is important to understand the intrinsic functioning of GANs. It also discusses the previous works done by researchers in the field of autoencoders which are the type of neural networks used to learn the efficient internal mapping of unlabeled data. Before diving into mechanisms of the GANs, it is efficient to grasp the related concepts in the field of unsupervised learning, which will enable the readers to understand the functioning of GANs. The image-to-Image translation is another use case of GANs, which will be the main focus in this chapter. In the following sub-sections, topics related to machine learning, supervised, unsupervised, and semi-supervised learning will be discussed along with reinforcement learning.

2.1 Machine Learning

In 1959, the American pioneer Arthur Samuel who was prominent in the field of Computer Gaming and Artificial Intelligence (A.I), invented the term Machine Learning (ML) [McCarthy 90]. The main concept behind this term is to teach machines to be independent in terms of functionality. Machine learning belongs to the sub-class of Artificial Intelligence in which scientists research and develop specific computer algorithms which refine their output through experience and data. This data is categorized as training data. A model in machine learning is built by computer algorithms based on sample data which is also known as training data, which is trained on sample data to predict or make decisions on unseen data without explicit help from outside. After many trials and errors, the model perfects itself to make better decisions unaccompanied.

2.1.1 Supervised Learning

It is known as a subcategory of machine learning where the labeled dataset is utilized for the training of the model. The result of this training is to allow the model to predict outcomes accurately or classify the unseen data. Initially, the accuracy of the model is very low however it gains confidence by improving the accuracy through loss function. Mathematically, it is written as an input variable (X), output as the variable (y), and the

function(F) is used to map from input to output i.e., Y = F(X). During the training, the supervised model attempts to minimize the loss function output by comparing it with real output and updating the weights and biases. This whole process is done through the Backpropagation technique. When the training is complete, the approximated mapping function is used to predict the output through unseen data. Supervised learning problems can be categorized into Regression and Classification problems. In classification, the output of the mapping function belongs to categories i.e., "CAT" or "DOG", "YES" or "NO". A regression problem arises when the mapping function outputs real values, which are predictions. There are notable weaknesses of supervised learning, and they must be considered before the final decision. For example, Computation time is huge or the time required for pre-processing of the data is very extensive. Overfitting can be easily achieved.

Important supervised algorithms are listed:

- Decision Trees
- K-Nearest Neighbours
- Neural Networks
- Naive Bayes
- Support Vector Machines (SVM)
- Linear and Logistic Regression

2.1.2 Unsupervised Learning

In this learning mechanism, unlabeled data is used for analysis and clustering. When the system does not have labeled data i.e., input data without any output. then unsupervised learning tends to find the pattern. This system is enforced to construct a better representation of its data and then generate inspired results from it. In the training process, an unsupervised model attempts to imitate the data and use the loss function to find the errors in an imitated output. Once the errors are found then the model tries to update its weight and biases to minimize the errors. As supervised learning deploy Backpropagation for the weights update, unsupervised learning utilizes algorithms including Boltzmann learning rule, Maximum likelihood method, Gibbs Sampling or backpropagating reconstruction errors, etc. As supervised learning has notable shortcomings, unsupervised learning also has a fair share of disadvantages. For example, it is impossible to get precise information about the underlying data. Less accuracy is also a problem because the training data has no labels. The model is required to do itself.

Important unsupervised learning methods are :

- Clustering
  - K-Means Clustering
    - Agglomerative clustering
    - Dendrogram

  - Hierarchical Clustering
- Association
- Principle Component Analysis (PCA)
- Neural Networks
  - Generative Adversarial Networks
  - Autoencoders

2.1.3 Semi Supervised Learning

Supervised and unsupervised learning has its weaknesses that can impact the outcome of the model. To label the data, researchers require a skilled human agent or physical experiment which increases the cost of huge datasets enormously. Unlabeled data tends to produce less accurate results, which renders the model infeasible. Due to these shortcomings, a new approach was conceived where there is a small percentage of the labeled data with a large percentage of unlabeled data during the training process.

2.1.4 Reinforcement Learning

This is a special branch of learning in which models force to acquire the commutative rewards for their sequence of decisions. Reinforcement learning extracts feedback from the environment to fix its path towards optimal decisions. This analogy can be further described as a game-like situation where an agent performs some actions while interacting with its environment. These actions are then either penalized or rewarded according to the set rules. The learning goal is to maximize the rewards and minimize the penalties.

2.2 Convolution Neural Network

A Convolution Neural Network (CNN) is a building block of GANs therefore it is very important to understand the basic concepts behind this algorithm. CNN is a deep learning algorithm that emerges from Neural Networks (NN). These Neural Networks are known as fully connected perceptrons similar to brain cells structure where each cell is connected to another cell. The main drawback of these networks is that they have a large number of parameters that generally fail to scale well. These Networks are very expensive to train in terms of computation power. Convolution Neural nets were born to compensate for the shortcomings provided by Neural Networks. Convolution Neural Networks allow disconnecting neurons according to the situation. They are also considered partially connected neural nets and they share weights. These special types of Neural Nets have seen remarkable success in the field of Computer Vision and Natural language processing. The most notable use cases of CNN are Image Analysis and Classification, Content Recreations and Recommendation Systems, etc.

The CNN is considered as a deep learning algorithm that takes multi-channel images and attache the learn-able weights and biases to the portions of an image which allows the model to focus on parts of an image instead of the full image. The CNN combines the deep

neural networks and Kernel Convolutions. These kernel convolutions are special matrices utilized for edge detection, blurring, sharpening, embossing, etc. The kernel is also known as filters which are normally hand-engineered however with enough training, a CNN can learn these filters. The main advantage of CNN is the ability to share the parameters weights of the particular feature map. This assists in the reduction of the parameters which eventually makes computation more efficient. The data pre-processing step is not very expensive as compared to other classification systems. After the advancements in deep learning, computer vision, and Compute power, CNN has observed huge success in accuracy and reliability.

A normal CNN architecture consists of the input layer, convolution layer, pooling layer, and fully connected layer. This architecture design is comparable to the connectivity pattern of human brain cells. The visual cortex is the main inspiration for the design of CNN's architecture. In the Visual cortex, the receptive field is the restricted area where neurons only respond to external stimuli. A simplified version of normal CNN architecture is visualized in 2.1

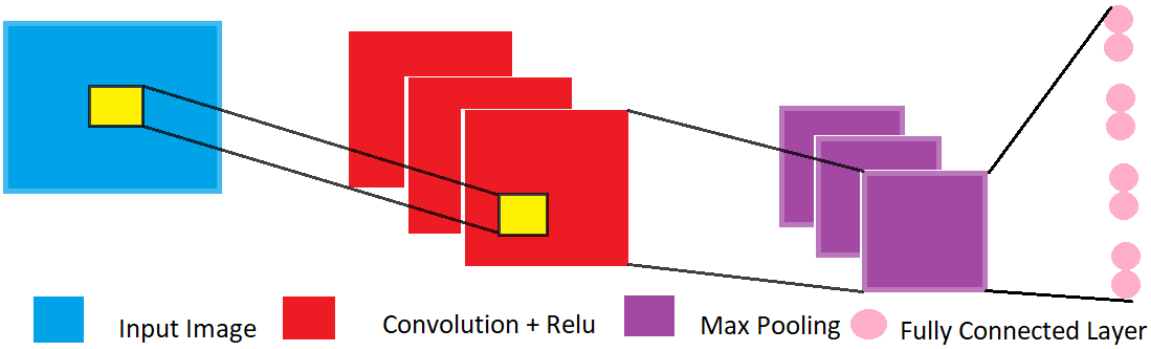


Figure 2.1: A visualisation of general architecture of CNN

- **Input Layer** - This is the first layer in any CNN model which is designed to get the image for the training and evaluation process. Images are consists of pixel values which are considered as neurons in Neural Networks.

- **Convolution Layer** - This layer is very crucial part of any CNN model. It consists of convolution filters or kernels which are simple mathematical operations on part of the image pixels. These filters create feature map which are repeated filter operations on the image. A feature map informs the strength and location detected feature in an image. The filter move through the image which is known as stride. A filter multiplies its numbers with pixel values and then perform summation to get the final output.

- **Pooling Layer** - Convolution layer produces the feature map which contains location and strength of detected feature. Pooling layer reduces the dimensionality of the produced feature map. The advantage of this layer is to reduce the complexity of model and lower the number of parameters which aides the computation. Max pooling , global pooling and average pooling are considered to be widely used pooling methods among scientists.

- **Fully Connected Layer** - In the fully connected layer, every neuron pf previous layer is connected to every activation unit of layer. In every model, the last layers are designated as full connected layer since they compile the data into final output. The main strength of fully connected layers is classification of input.

2.2.1 Activation Units

In CNN, Convolution layers perform convolution computations on the input layer which introduce linearity in the network. Activation units add non-linearity in the network by performing a weighted summation of all the neurons from the previous layer and adding the bias. After this operation, the output value can range from -inf to +inf, which is not a good decision for the neuron to be active or not. By studying the human brain, scientists have observed that neurons can generate stimulus signals when the input signal crosses a certain threshold. A similar idea is imported into CNN by using activation functions. If the output of the neuron crosses a certain threshold then this neuron will be fired up to produce input for the next layer. The simple activation function will output value if the input crosses a certain threshold however, this simple case can create issues when the problem is binary classification.

Figure 2.2 shows the visual comparison between sigmoid and Tanh.

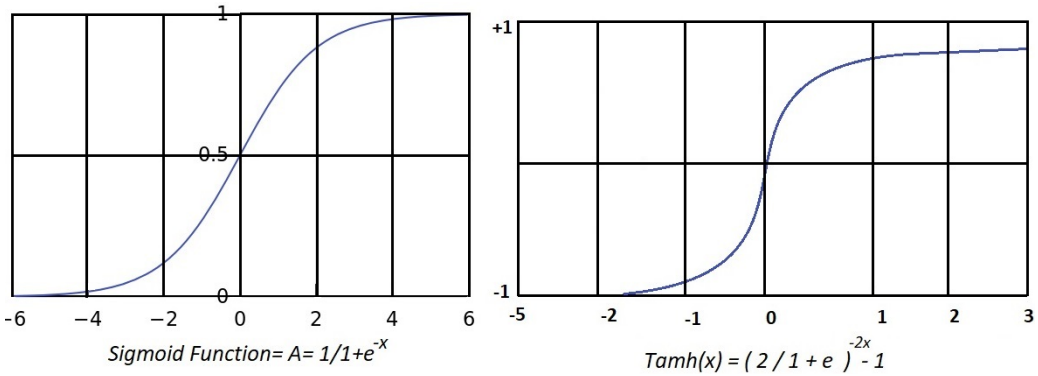


Figure 2.2: A Visual comparison between sigmoid and tanh function

- **Sigmoid Function** - This is a non-linear activation function that takes input and produces the output in the range (0,1). The most intriguing problem in this activation function is vanishing gradient issues near the horizontal area of the function line. In this area, networks refuse to learn.

- **Tanh Function** - This function has a striking similarity with the sigmoid function. The range of Tanh function is between (-1,1). Like sigmoid, Tanh also suffers from vanishing gradient problems however it has feasibly stronger gradients.

- **Rectified Linear Unit** - This function is also known as Relu and mathematical formulation is f(x) = max(0,x). Relu is also non-linear. The range of Relu is (0,+inf) which means the negative part of the output is strictly restricted to propagate through the Network.

- **Rectified Linear Unit** - In the ReLu, vanishing gradient is the issue due to restriction on the negative component of the output. Leaky ReLu allows small gradient for negative input which helps to curb the vanishing gradient issue.

2.2.2 Training and Evaluation

In supervised learning, the CNN model requires training datasets that consist of input data with output labels. A training process consists of two phases, the forward phase, and the backward phase. In the forward phase, input data simply propagates through the network once, and in the backward phase, gradient. During the training process, a loss function can calculate the difference between model-generated output and actual output, and consequently, this difference is propagated back to the network for the adaption of weights. This is known as the Backpropagation algorithm for CNN in the supervised learning setting.

In the backpropagation Algorithm, every layer gets a gradient of the loss concerning its output after this operation, each layer returns the loss's gradient concerning input. These gradients are computed very efficiently with backpropagation which allows the researchers to use gradient methods for the training of multi-layers of CNN, one method is called Gradient Descent, or a variant such as stochastic gradient descent. Gradient Descent (GD)

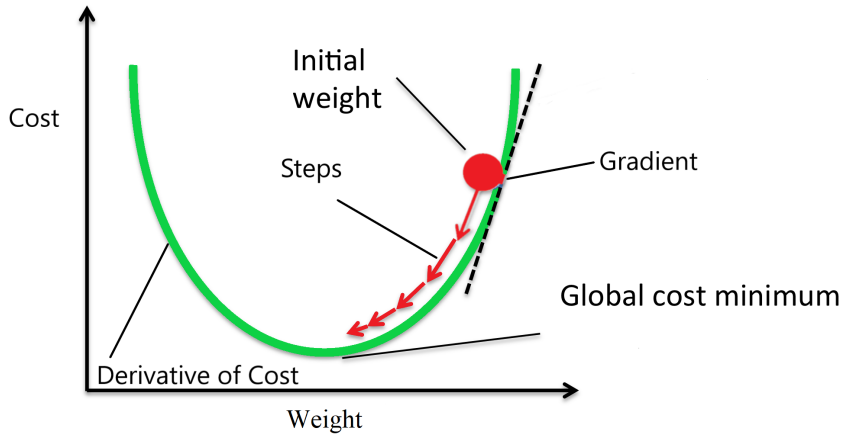


Figure 2.3: A visual 2D depiction of gradient descent Algorithm against cost function

is an iterative method that is primarily used to find the local maxima or minima of the function. The main benefit of the GD method is to minimize the loss function. Not every function can be used for the GD algorithm because the function needs to be differentiable and convex. Differentiable functions have derivatives for each point in their domain. After the initialization of starting point, the GD algorithm, at the current position, calculates the next point and scales it with the learning rate. This is known as the scale step. It then subtracts the output to minimize the function. The GD algorithm must converge the function to find the value of weights which gives minimum error. When the model converges after many iterations, it means the error is substantially low.

Evaluation is a very important step in the aspect of any A.I related project. When the CNN model gives a satisfying accuracy score against the unseen data then it is considered as a production-ready well-trained model. There are many metrics available that test

different strengths of the CNN model. The selection of the evaluation metric comes from the use case of the CNN Model. Some of the famous evaluation metrics are listed below

- Classification Accuracy
- Logarithmic Loss
- Confusion Metrics
- Mean Absolute Error
- F1
- Area Under Curve (AOC

2.3 Generative Adversarial Networks

Generative Adversarial Networks (GANs) belong to an unsupervised class of machine learning paradigms. It consists of two networks that are competing with each other for the Nash equilibrium. One network known as a discriminator has a job to distinguish between real and fake data whereas another network known as the generator has a job to generate fake data as much as similar to real data. These two networks are playing an adversarial game against each other in which the generator can be viewed as a counterfeiter and discriminator as a detective. In the first turn, the generator generates fake data from latent space or noise and the discriminator compares the fake data with real data and labels the data from the generator as fake since the loss is huge. After many iterations for training, it is very difficult for the discriminator to distinguish between real and fake data. The fig2.4 depicts the simple architecture GANs for the viewers.

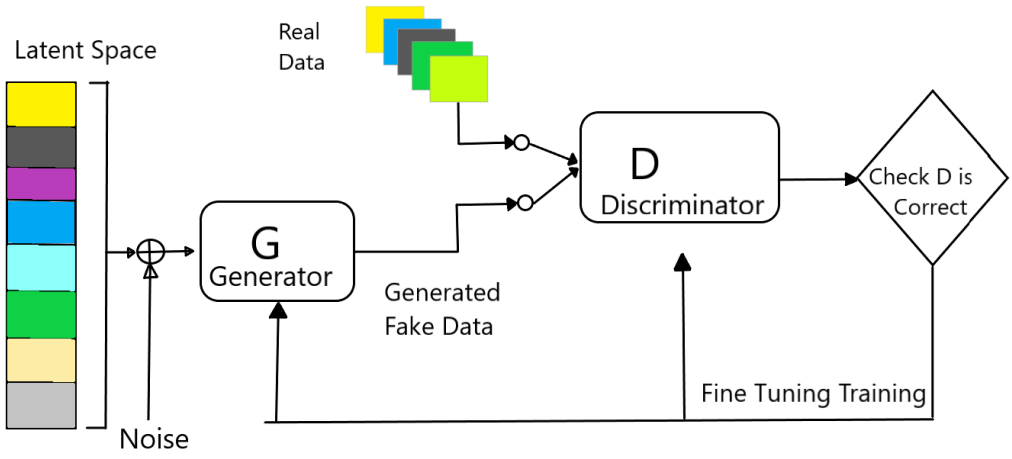


Figure 2.4: A visual representation of the GANs

The observed adversarial framework performs collectively superior when the corresponding networks are multi-perception or CNN. The loss function of Gans is the combination of the discriminator's loss and the generator's loss. It is important to discuss the Generator part in the GANs to understand the complete functionality.

2.3.1 The Generator

The generator is a network in GANs that has the sole purpose to create fake data samples by integrating the response or feedback from the discriminator. It also trains to learn to deceive the discriminator by making its data classified as real. Initially, the generator network takes the random input from latent noise and converts it into data point x. This data point x is input to the discriminator network which performs classification between real and fake data instances. Input noise is defined by $P_z(Z)$ which corresponds to distribution P_g of the generator over given data x. Generative models have the data distribution and also the ability to provide information about the likelihood of the data belonging to the same distribution. It captures the joint probability $P(X, Y)$ if there are labels Y otherwise simple $P(X)$ [Goodfellow 14].

The generator must imitate the distribution of the data to deceive the discriminator. For example, The noise vector and class label Y is given to the generator, which consists of a neural network. This model produces features $G(z)$) which it believes belongs to data distribution. Generator models, the probability of feature given the class label is present.

A neural network requires some input to start the forward pass propagation. In normal conditions, the neural network is fed with the input data that needs to be classified. However, in the case of GANs, neural networks produce completely new data instances which belong to the noise data distribution. Therefore, it is important to expose the network with noise which allows generating a varied variety of data instances. However, Researchers have concluded that nose vector is not important as it is feasible to choose data points from any uniformly distributed data. It is a known fact that the space from which noise is samples has smaller dimensions than the dimensionality of the generated data point. In comparison to the discriminative model, a generative model must model the feasibly difficult data distribution. These data distribution or very complex to capture as generator tries to model how the data is placed in distribution space.

Before diving into the mathematical formulation of the GANs network, it is important to define some parameters and variables. These variables and parameters will make it easy for the readers to understand the formulation.

- D = Discriminator
- G = Generator
- z^i = Random Noise
- x^i = Real Data Point
- θ_d = Parameter for Discriminator
- θ_g = Parameter for Generator
- $P_z(Z)$ = Input Noise Distribution
- $P_{data}(x)$ = Original Data Distribution
- $P_g(x)$ = Generated Distribution

2.3.2 Generative Loss

The goal of the generator's loss is to penalize the network for failing to deceive the discriminator. In the below given equation, it is observable that objective function is achieved through

$$G(z : \theta_g) = log(1 - D(G(z)) \tag{2.1}$$

The total cost of overall data points can be written as

$$G(z : \theta_g) = \frac{1}{m} \sum_{i=1}^{m} log(1 - D(G(z^i))) \tag{2.2}$$

In the above equation, generator $G()$ is using random noise z^i to produce the data point which it thinks belongs to real data distribution and the discriminator performs the classification on the generated out from $G()$. $D(G(z^i))$ is the discriminator approximation of the probability that fake data generated from $G(z^i)$ is either belong to sample data distribution or from latent space. Take log of $1 - D(G(z^i)$ and in this setting the discriminator produces either 0 or 1. Optimally , it is required from discriminator to produce 0 which results in the $log(1) = 0$. If the generator is able to deceive the discriminator then this loss of the generator will be close to 0. It is also noted that the inventor of the GANs changed the generator's loss to $G(z : \theta_g) = log(D(G(z^i)))$ for better stability and reduction of saturation problem. The generator wants to minimize the above equation in order to fool the discriminator.

2.3.3 The Discriminator

This network is designed to differentiate between real data from real data distribution and generated data from noise distribution. In this mathematical formulation, the discriminator is defined by $D(x, \theta_d)$ that produces single scalar output. In theoretical analysis, $D(x, \theta_d)$ and $G(z, \theta_g)$ must be given enough capacity to recover from training criterion. The discriminator can be designed as any neural network which must classify the real data input as real and fake data input as fake. The objective function of the Discriminator is $D(x, \theta_d)$ must be maximized and $G(z, \theta_g)$ must be minimized. Both networks produce a classification score between 0 and 1, it is important that the discriminator maximizes the probability estimation of the real data while minimizing the probability estimation of the fake data.

2.3.4 Discriminator Loss

Discriminator $D(x, \theta_d)$ loss function determines the performance of the network against the generated output. From the discriminator's perceptive, it must distinguish between real and fake data. For the classification of real data, the discriminator produces 1 and for the classification of the fake data, it produces 0. In the below equation, it is observable that the equation of discriminative loss is divided into two sub equations where $D(x, \theta_d)$ deals with the probability estimation of real data x and $D(z, \theta_d)$ deals with probability estimation of fake data z generated from the generator.

$$D(x : \theta_d) = log(D(x)) \tag{2.3}$$

For fake data point z

$$D(z : \theta_d) = log(1 - D(G(z)) \tag{2.4}$$

Combining both equations gives

$$log(D(x) + log(1 - D(G(z))) \tag{2.5}$$

The total cost of all the instances of data distribution in final discriminator's loss function looks like this.

$$D(x : \theta_D) = \frac{1}{m}\sum_{i=1}^{m}[logD(x^i) + log(1 - D(G(z^i)))] \tag{2.6}$$

It is observable from the above equation that in $logD(x^i)$ discriminator$D(x^i)$ must produce probability estimation 1. This means that from discriminator's perspective, the real data from data distribution x^i is classified as 1 which is computed as $log(1) = 0$. In the other part of the $1 + log(1 - D(G(z^i)))$, the generator $G(z^i)$ produces the fake data from the noise z^i and discriminator must produce 0 because this data point belongs to noise. The final values in this portion of equation looks like $log(1 - 0) = 0$. If the generator can deceive the discriminator then $D(G(z^i))$ will produce some value either closer to 0 or 1. The discriminator wants to maximize the above equation. However, from the generator's perceptive, the generator wants to minimize the generator loss function which means it wants to deceive the discriminator.

Now discriminator D and generator G have their respective loss functions and they play the two-player min-max game with the value functions $V(D, G)$. The gradients with respect to their parameters are calculated for G,D and propagate through networks independently. This is the optimization problem where it says find G that minimize and find D that maximize the $V(D, G)$ cost function. The equation of value function $V(D, G)$ for a single data point is written like this:

$$Min_G \;\; Max_D V(D, G) = E_{x \in p_{data}(x)}[logD(x)] + E_{z \in p_z(z)}[log1 - D(G(z)] \tag{2.7}$$

where Value function of Discriminator $V(D)$ written like this:

$$Max_D V(D) = E_{x \in p_{data}(x)}[logD(x)] + E_{z \in p_z(z)}[log1 - D(G(z)] \tag{2.8}$$

and the Value function $V(G)$ written like this :

$$Min_G \; V(G) = E_{z \in p_z(z)}[log1 - D(G(z)] \tag{2.9}$$

In the above equation, the objective function is to optimize G in order to fool the discriminator. However $E_{z \in p_z(z)}[log1 - D(G(z)]$ has very weak gradients which do not converge well in then so in practice it is recommended to maximize the Generator G loss function. This loss function written like $Max_G E_{z \in p_{data}(z)}[logD(G(z)]$ which produces non-saturating gradient which is better for training. There are different types of loss function available which has varied strengths and weaknesses. The GANs try to imitate the probability distribution of the training data therefore they should use loss functions that

can guide them towards an optimal solution. The general idea of these loss functions is to calculate the distance between the distribution of real and the distribution of generated data. The mathematical statement is presented now and in order to solve this optimization problem, a training algorithm must be devised. This training algorithm must find an optimal solution of $V(D, G)$ to generate real data points where the discriminator can not distinguish between real and fake. This is achieved through extensive unsupervised training which will be under discussion in the next section.

2.3.5 Training and Evaluation

GANs are consist of two separately trainable networks. These networks can deploy any architecture in order to achieve their desired tasks however training these networks requires special care in order to achieve optimal results. GANs training requires addressing two major complications. The first complication is to manage two different networks in the training process. The discriminator network needs to train for one or more epochs since it must classify the read data point and fake data point. Both of these networks need to train parallelly and convergence is significantly complicated to identify. It is important to keep the generator constant during the discriminator training since it has to distinguish between real and fake data and must also determine the generator's flaws.

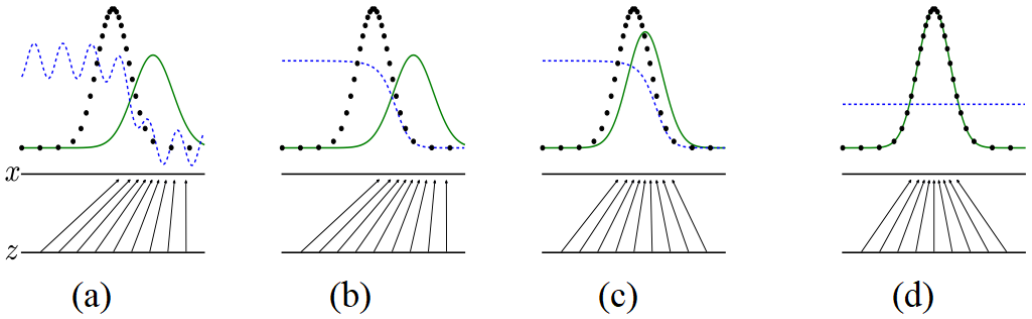


Figure 2.5: A Visual representation of training of discriminative network D and generative network G where D is represented as dashed blue and real data distribution is represented as block doted line. Green solid line represents generative data distribution P_g

In the Fig2.5, It can be observed that GANs are trained parallelly by updating the parameters of discriminative data distribution (Blue dashed line) which allows D to distinguish between sample points generated from real data distribution (black, dashed line) $P_{data}(x)$ from the data products from generative distribution $P_G(x)$. The z is data generated from latent space which is visualized as a lower horizontal line in the above figure. The above second horizontal line is represented the data x from real distribution. The arrows from the lower to upper horizontal line are suggesting about the mapping $x = G(z)$ forces the non-uniform distribution P_g. The generator network G starts from the higher region of generated data distribution and moves towards lover density of distribution. In the above graph (a), G graph and real data distribution graph have significant similarities which means the generator network is mapping real data distribution very nicely. However, D is a marginally better accurate classifier. In the second setting (b), D is trained better to distinguish between real and fake data as it follows the black dotted line and finally converges to $D^*(x) = \frac{P_{data}(x)}{P_{data}(x)+P_g(x)}$. After the D converges, the gradient of D pushes G

to move to a region that is more likely to be classified as data. In graph (c) After many epochs of training, G and D have converges nicely because $P_g = P_{data}$

For the training purpose, it is feasible to use mini-batch stochastic gradient training for the GANs as it converges better when trained on small batches. The complete process of the training algorithm is defined below. K is a hyper-parameter and can be selected according to the availability of computation resources.

- **for** each training iteration **do**
  - **for** k steps **do**
    - Sample the m noise data samples $z_1, ...z_m$ generated through generator network
    - Sample the m real data samples $x_1, ...x_m$ from real data distribution. it is important to associate the label 0 to fake data point and 1 to real data point.
    - Pass these data samples to Discriminator network and update the parameters by ascending the gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [log\ D(x^i) + log\ (1 - D(G(z^i)))]$$

    - Compute the gradients w.r.t discriminator
    - **end for**
    - Sample the m noise data samples $z_1, ...z_m$ generated through generator network.
    - Pass these data samples to Discriminator network and update the parameters by descending the gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} log\ (1 - D(G(z^i)))]$$

    - compute the gradient w.r.t generator
- **end for**

Many issues arise with the training of the GANs as two networks need to train separately. All of these issues are now the area of active research and none of them have completely been solved. Vanishing gradient is the common problem that arises when the discriminator is very optimized to classify the data. This forces generator to fail. An optimal discriminator network does not share substantial information to the generator for the progress. A balance must be set between these two networks. To solve this issue, Wasserstein loss is designed to prevent the vanishing gradient. Modified mini-max loss is also another solution for this specific problem. In this solution, it is recommended by the researchers to modify the generator loss to maximize the $logD(G(z))$

Mode Collapse is another serious problem that arises when training the GANs. The main objective of GANs is to output a wide variety of data by learning to map the real data distribution. For example, optimally GANs must generate different face examples for

each random input. However, there are instances when the generator network outputs the same data point for each random input, which means it has learned to produce an only specific output. Theoretically, the generator has learned to produce the same data that the discriminator classifies as real. When the generator network starts producing similar data points of the set of data points constantly and discriminator networks learn to reject these similar inputs then the problem of mode collapse arises [Thanh-Tung 20].

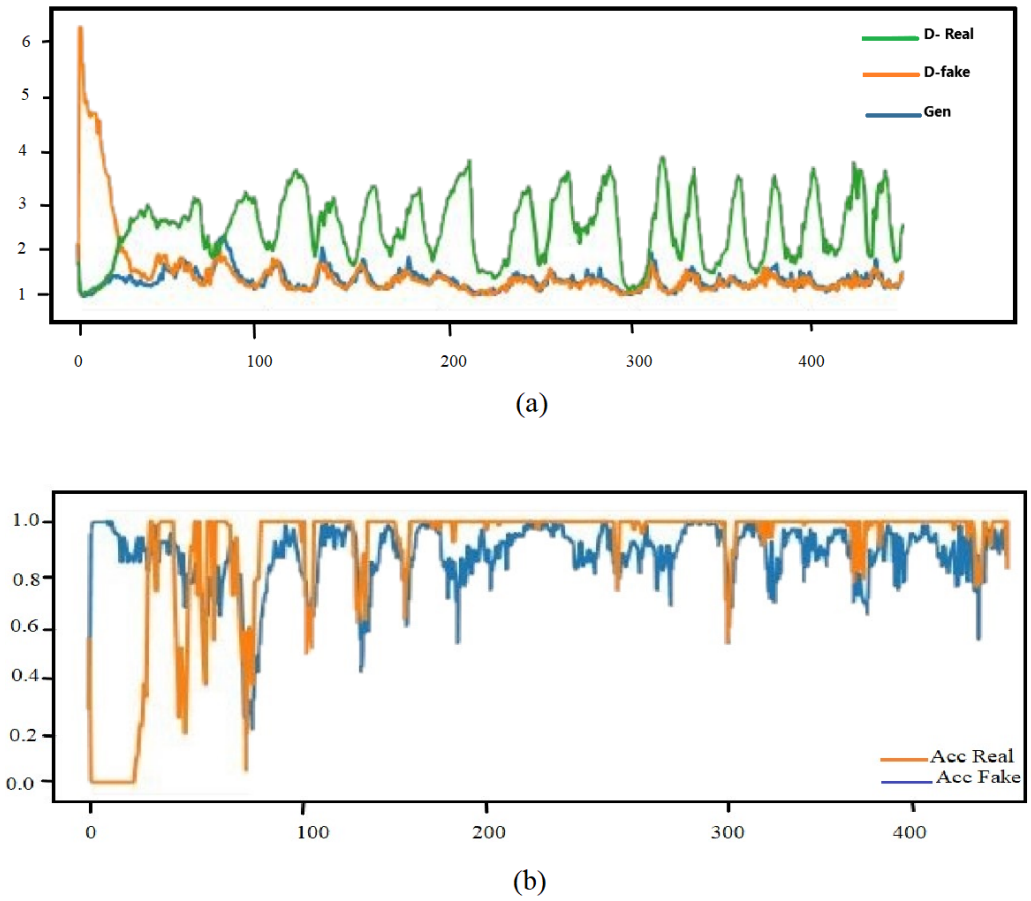


Figure 2.6: A graphical representation of the mode collapse in GANs. In (a), blue graph from generator(G) is similar to discriminator(D) graph (Orange) which means G is not learning to defeat D. In (b), the Accuracy of real images (orange) is not different from accuracy of fake images. the part (a) is from generator's perceptive the part (b) is from discriminator's perceptive

During the training, it is helpful to generate the graphs of D and G which can show insightful information. In the above Fig 2.6(b), it can be visualized that in the entire run of the training, the discriminator ability to classify between real and fake data points is very high. This suggests that the generator network has not learned enough to produce data points which discriminator can not reject or the generator network has been producing poor quality data points which are easily rejected by the discriminator's network.

2.3.6 Evaluation of GANs

GANs are complex architectures that are trained to produce fake data points from the real data distribution. Every other deep learning network is trained with the loss function until

it converges. However, in GANs architecture, a generator's loss function is the discriminator network. This network guides the generator to train itself to produce identical data points. It is important for the stable training that both networks initial from zero states and are trained together to sustain the equilibrium. As stated earlier the generator does not have an objective loss function used to train and there is no way to assess the progress of the training or the quality of the output but only through discriminator. Therefore, researchers have developed some techniques to assess the quality of the data points produced by GANs. However, the objective evaluation of GANs remains in the active research area. The evaluation metrics of GANs are divided into two categories: the first is Quantitative Evaluation techniques and the second is qualitative Evaluation techniques. In qualitative measures, there is an element of human subjective evaluation which involves manual inspection. In this process, human participants are asked to evaluate the pair of images without prior knowledge. The main drawback of these techniques is they are labor-intensive tasks and human's ability to evaluate, is also subjective. In quantitative measures, a calculated numerical score decides the quality of the generated output. The process of calculation varies because each measure has its strength and weaknesses. There are many quantitative measures available for the evaluations[Barua 19]. The most famous ones are listed below.

- **Qualitative Measures**
  - Rapid Scene Categorization
  - Evaluating Mode Drop and Mode Collapse.
  - Nearest Neighbors.
  - Investigating and Visualizing the Internals of Networks.
- **Quantitative Measures**
  - Inception Score (IS)
  - Modified Inception Score (m-IS)
  - Frechet Inception Distance (FID)
  - The Wasserstein Critic
  - Generative Adversarial Metric (GAM)
  - Geometry Score

In this project, Frechet Inception Distance (FID) is used for the evaluation of the images generated through GANs. FID is a metric that calculates the distance between feature vectors from real and generated images. A lower score means that generated images are closer to real images in terms of features [Heusel 17]. In the later part of this thesis, a detailed discussion will be done on this technique.

3. Related Work

In recent years, the machine learning community has experienced a surge in the advancements of GANs. Many different variants of adversarial networks have been designed and developed through extensive research. From those variants, image-to-image translation is under observation for this project. As stated earlier, the core objective of this thesis is to utilize synthetic images for the style transfer using cyclic GANs. To study the cyclic Gans, it is very important to understand all the related knowledge leading up to image-to-image translation technology.

3.1 Traditional Feature Extraction and Reconstruction

In classical image processing, there have been many techniques developed to extracts the features from multi-channel images and reconstruct them for enhancement or manipulation purposes [Balan 18]. In a given image, the feature set is subdivided into color features, texture features, and shape features which provide variables to manipulate the image space. There are many traditional techniques to reconstruct these feature set in an image however machine learning-based feature extraction techniques have surpassed the previous techniques in quality and quantity. These traditional techniques are Geometric transformations where mathematical transformations are deployed to scale, rotate, transpose or change the image array for the production of the modified image however these transformations significantly degrades the quality of the output images. Another famous technique is known as a back-projection which states that the distribution of image histogram is utilized for the reconstruction of the pixels which map the similar distribution. In another computed tomography reconstruction technique, the Fourier slice theorem says that if there is objective function f(x,y) in x,y coordinates and there is projection function p(x,y) then after applying 2D Fourier transform, the system produces its counterpart in Fourier domain. In practice, it is unclear about the objective function but projections of this objective function are known due to the line integrals. it is stated in this theorem that 1-D Fourier of the projection function, is also equivalent to the slice in the Fourier 2D domain. Theoretically, it is possible to reconstruct any object y just acquiring different projection[Umbaugh 10].

3.2 Reconstruction through Autoencoders

Before the advent of GANs, scientists were searching for different deep learning-based techniques where the mapping of data is learned and re-used again for the reconstruction of the image but with probability distribution. In 1986, a few researchers [Rumelhart 85] published their work in which they presented an idea to design a specific type of system of neural networks in which it can encode the input data into significant representation and then decode it back in such a way the reconstructed output is as similar as input. This system is known as autoencoders. In other words, autoencoders learn to recreate their inputs through unsupervised learning techniques. As it can be observed in Fig 3.1 the architecture of autoencoders consists of two neural networks where one is called an encoder, which learns to compress the original input data into encoding while the other is called a decoder which learns to decode the encoding from encoder into original form. After further inspection, it is revealed that autoencoders have a bottleneck in the network, which helps to compress the knowledge representation. When the features in the given data have no correlations or there are independent of each other then it is significantly difficult to reconstruct that data. However, if there exists any form of correlations between the features then autoencoders networks can learn the structure and reconstruct it through the bottleneck.

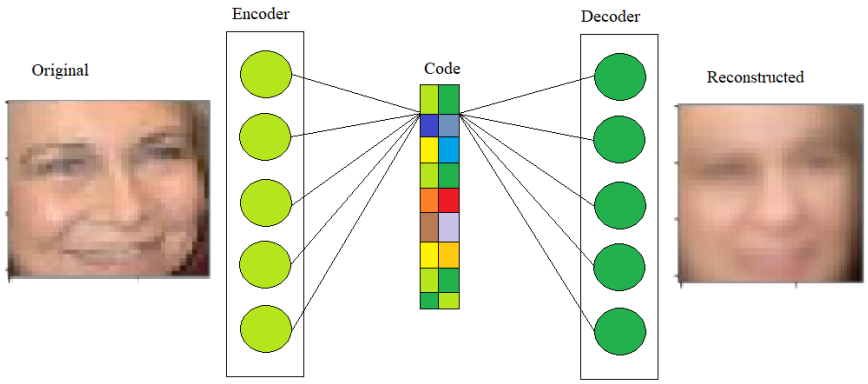


Figure 3.1: A visual representation of the autoencoders where input image is encoded through internal coding function.The decoder extract the information coding function for the reconstruction of input image.

The encoder in the autoencoders consists of artificial neural networks(ANN) or CNN, which can search the minute data representation or features and then encode this information in a compressed manner. The decoder is ANN or CNN which learns to read the encoded compressed data. During the training, the encoder objective is to minimize the mean square error(MSE) loss while reconstructing the images. The error from the decoder is backpropagated through the network for the updation of all the parameters. Evaluation of the decoder and encoder allows learning the parameters to become better in the reconstruction of the images. There is a major limitation attached to the autoencoders which do not allow them to be good generative models for new content generation. After the proper training of autoencoders, it is still not feasible to generate new content with the same probability data distribution. Even if the latent space is well organized by the

encoder during the training process, any point can be taken from this space and the decoder will work as a perfect generator for the generation of new data. However, it is very grueling to achieve because it depends on the distribution of the data in input space, the architecture of the encoder. Encoders have a very high degree of freedom to generate encoding of the data in such a way that decoder can reconstruct it without any loss. they have a natural tendency to overfit the learning due to their architectural design. This leads to severe overfitting which is not feasible for the generation of new data.

Variational Autoencoders (VAEs) are another type of generative model in which the distribution of the encoding is regularized during the training process. This regularization allows better latent space representation therefore new data can be generated from this space. The training of the VAEs is regularized to avoid overfitting which ensures that latent space has good properties for the generation of new content or data. Due to the complexity of the theoretical base of VAEs, researchers have chosen GANs since they provide more generalization in terms of data generation. They can learn the mapping from nose data sample to complex target data sample which solves the problem of the availability of training data [Doersch 16].

3.3 Image-to-Image Translation

After the advent of GANs[Zhao 17], researchers have achieved significant results in image generation, which allows them to create a different image from the training data. This generalization gives the ability to create new data from the probability distribution of input data space. The participants of this research [Denton 15] introduced a generative model proficient in the generation of high-quality images from the distribution of natural images. This model utilizes a Laplacian pyramid framework in which a cascade of CNNs is trained to generate images in a coarse-to-fine manner. There is a separate GANs model at each level which is designed to capture different coefficients of natural images. From the lowest level, GANs generate images from noise samples. This first image generation is of very low quality however this image is input data for the above level GANs. This work

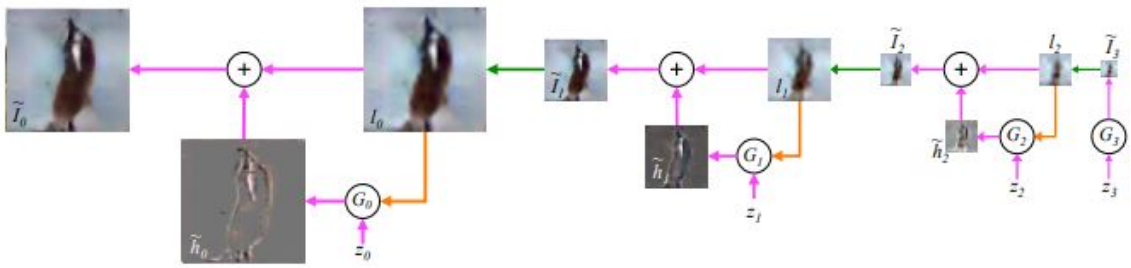


Figure 3.2: This is visualisation of training procedure of LAPGAN Model where Z_3 (right side) is the noise vector which is input to G_3 GAN then for the next model G_3, conditional variable (orange) is used. This method repeats for two levels until it produce the final results.

has its limitations however it produces quality results. It is not possible to cover all the data distribution as there is the possibility that this model is assigning a low probability to the portions of natural images.

Image editing has utilized the power of GANs[Zhao 17] to produce natural-looking images from the natural image data distribution. It has been shown in this work that images can

be manipulated with artistic sense to create real images. A portion of the image is selected before the training which is part of the dataset. In this way, it is possible to control the generation of the data through constraint optimization in GANs. First, researchers have projected the image onto lower-dimensional latent vector representation, then regenerate the same low-dimensional image for the manipulation in color or shape space.

The first idea of image-to-image translation was given by Hertzmann et al.'s Image Analogies [Hertzmann 01]. In this work, researchers designed framework which consists of two stages: a design phase, in which paired images, one is in its original form and second image is filtered version of first image, is collected as training data, and an application phase, a trained filter from the training data or paired images, is convoluted to unseen target image to get the similar filtered results. Texture synthesis process[Efros 99] was deployed in which from single pixel image texture is grown to new image using Markov random field model and conditional distribution of pixel's neighbors is synthesized.

Many other approaches have been seen significant quality results when input-output datasets have been used by CNNs to learn the parametric translation function. The CNNs can extract hidden hierarchies of features in a given image. When these CNNs are trained end-to-end and pixel-to-pixel, it has been observed that they can exceed state-of-the-art segmentation in terms of quality. Researchers have shown that when fully convolution networks are given input data with any random size, they produce the segmented image of the same size.

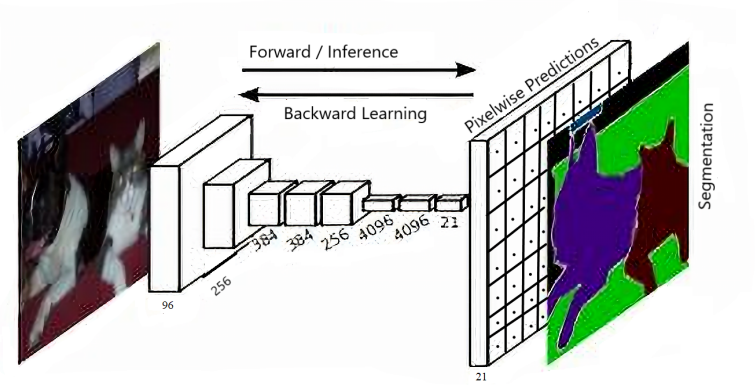


Figure 3.3: A visual representation shown for per pixel segmentation is achieved through fully convoluted networks after end-to-end training.

Full convoluted networks are specialized in the prediction of spatially dense data in a given image. Then other state-of-art deep neural networks are deployed for the per-pixel classification through transfer learning of trained representation. In this given architecture, researchers designed a skip sub-architecture that adds the semantic information of the pixel and appearance information to get the final segmented image. It can be observed that image-to-image translation is very possible in these settings as the input image is converted into the fully segmented image however generalization is not possible. It is not possible to create new unseen data from the existing training data. GANs are specially designed for the generation of new data points which lie within the probability distribution curve of the training data.

Isola et al. [Isola 17] presented a state-of-art framework that produced very good quality image-to-image translated data. This framework is called pixel2style2pixel (pSp) which contains an encoder and GAN generator. The encoder is designed to generate the style vectors which are input to the style GAN generator for the creation of extended latent space. Latent space is the space in which similar data are closer and different data are far in distance to each other. In this paper, this latent space is specifically referred to as $W+$ space. Style vector belongs to the vector art shapes which includes points, lines, shapes, and curves computed through mathematical formulas. Researchers have shown without the need for special optimization, an encoder can implant real natural images into $W+$. The encoder is also designed to encode from input space to latent space in such a way that it can solve all objectives related to image-to-image translation. The notable feature in this architecture is that encoders can achieve all the objectives without the presence of stylized images. It has been shown that the training process is simplified if StyleGan is used because there is no adversary network. Therefore, solving pix-to-pix connections is feasibly solve able.

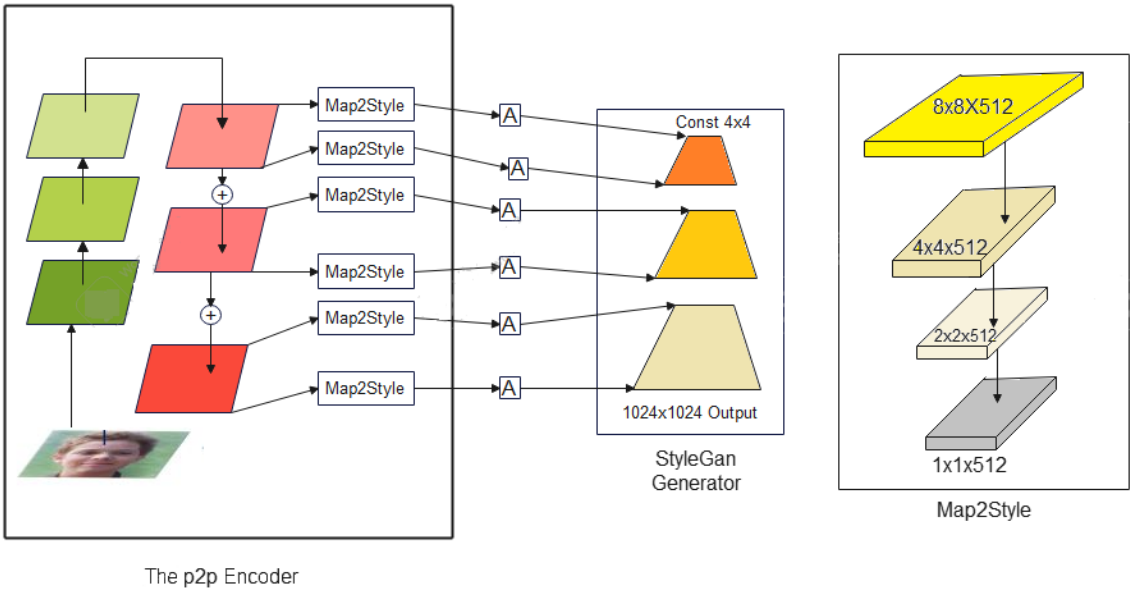


Figure 3.4: This is the visual depiction of pSp architecture.By using the res-net backbone, features maps are extracted then small mapping network is trained to extract the style vectors from corresponding feature map.

In the above Fig 3.4, the researchers have utilized an encoder as the backbone of the architecture, with a feature pyramid. This encoder generates three levels of feature maps through which styles are extracted with the help of the map2style network. These styles are direct correspondence to the hierarchical representations, and they fed into the generator for the reconstruction of the image. This whole architecture produces output results from input pixel to output pixel with the intermediate style representation, giving generalization.

This p2p architecture produces feasible quality results however there are limitations attached to this methodology. It fails to generate the complex structure in an image due to the complexity of the encoder. For the generation of high-quality images, it must be

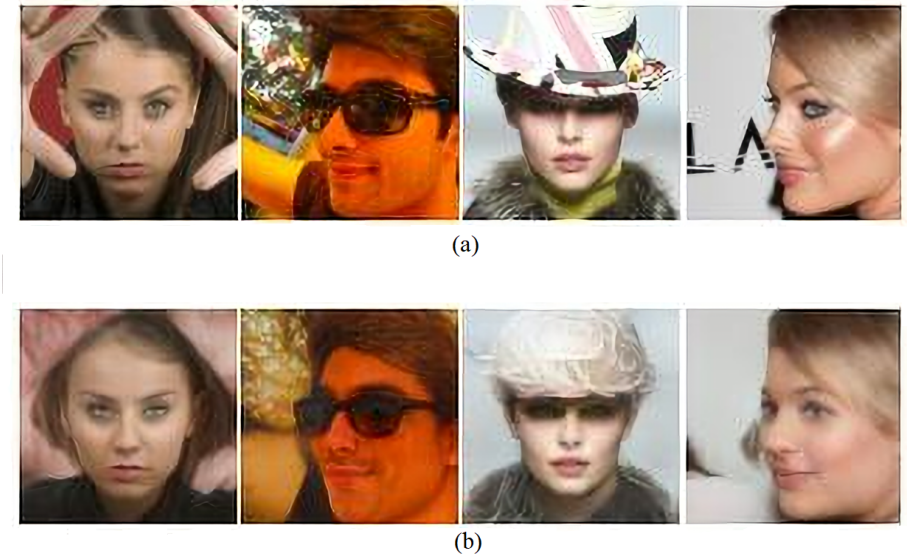


Figure 3.5: This is the visual representation of the complex tests cases for p2pStyleGan methodologies in which GANs are finding it difficult to differentiate between background and foreground. The (a) is real images and (b) is generated images.

considered that pre-trained StyleGan is utilized for translation tasks. This p2p architecture struggles to produce finer details of global classes of images that contain complex structures. This architecture also fails to preserve the finer details of the background of images as it can be observed in the examples in Fig 3.5.

In all of the methodologies mentioned in this thesis, one of the important requirements is to have paired dataset however this project needs to deal with unpaired images in the training process. To explain this furthermore, A GAN must be able to generate the images from the input domain to the target domain while keeping to style preserved during the translation. Rosal et al.[Rosales 03] purposed an unsupervised approach for the image synthesis using the Bayesian framework for inference. The researchers extracted patches based on the Markov random field from the source image $P(X)$ and to calculate the likelihood of $P(y|x)$, a Bayesian network is deployed. These filtered patches can be applied to images belonging to any domain which gives a sense of image translation. As stated earlier, this methodology does not require paired image dataset to produce an image to image translation as it relies on a computationally efficient probabilistic inference algorithm. In the above Fig 3.6, it can be observed how patches based on Markov random field can be applied from source image to target image However this methodology lacks generalization which means new data can not be generated through this technique.

It has been recently observed that utilization of GAN for the generation of high-quality images Keras et al. [Karras 19] presented the style generative adversarial network, also referred to as Style GAN which is an extended version of the basic GAN. In this version, the researchers have implemented extensive changes to accommodate the style transfer from source to target domain with persistent quality. A mapping network is designed to establish corresponding connections between latent space and intermediate latent space. It is evident from this methodology that deploying intermediate latent space to control style at each pixel generated by the generator network, with the addition of noise as a source of

Figure 3.6: The image (a) is Starry Night by Von Gogh's and Stylization is transferred to source image (c) through unsupervised probabilistic inference

variation, produces high-quality stylized images. In general, researchers believed that by improving the discriminator in GAN, it is possible to improve the training of the generator which will result in high quality and high variance in the synthetic image generation. However limited studies about the internal working of the generator have resulted in the construction of low-quality images. There are many ways to control the different properties of generated images. These properties can be in color space, shape space, or in style space which can have significant effects on higher properties like background, foreground, or finer detail of presented object in images.

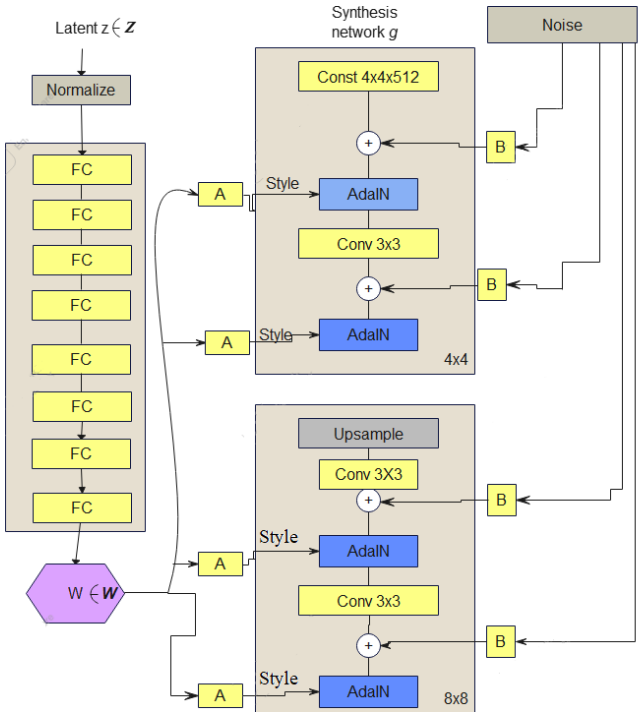


Figure 3.7: This is the visual architectural representation of the style GAN. The generator (left column) is trained to map the input data to intermediate latent space W then it guides the generator through Adaptive instance Normalization (AdaIN) at every convolution layer. This intimidated input space data is then given to synthesis network for the generation of final stylized image.

Style GANs is an extended version of specialized GANs known as Progressive growing

GANs(PG GANs). From this special GAN, researchers can generate very high-quality and high-resolution images without any introduction of artifacts or irregularities. This is achieved by a gradual expansion of generators and discriminators network from generating small to high-resolution images, in the training process. Style GANs borrow the same methodologies to generate high-resolution images. During the training process, style GANs change the generator network's architecture noticeably to mimic the incremental growth of Progressive GANs. There are two sources of randomness, which are utilized for the reconstruction of synthetic images: mapping network and layers of noise. The mapping network produces a vector that outlines pixel-wise interrogated style in the generator network. This is achieved through a layer that is named adaptive instance normalization (AdaIN) and it gives control over the generated image's style. The styleGAN architecture consists of five modifications that are added and evaluated gradually during the training. These incremental changes are Baseline Progressive GAN, Inclusion of tuning and bilinear up-sampling, Addition of AdaIN(styles), Removal of latent vector for generator, the addition of noise, and regularization for each block.

Style GANs are designed to be task-specific which means they are not general purpose. They produce high-quality results of fake faces while in training, it must be clear where the eyes, nose, and face pose is. They also require significant computation power to complete the training. Due to model complexity, there are high chances of unstable training which can be problematic for less complex use cases. Style GANs also exhibit the droplets like a blob in the generated images which can be infeasible when generation natural images like plants, trees. It has been observed from All variants of the GANs that they are designed

Figure 3.8: It can be observed that style GANs produce artifacts in the generated images.

to handle different tasks. Some extensions of GANs produce high-quality results of Facial Data and some are good with natural images. It has been also seen that GANs require huge computation power to generate data with style transfer which can be expensive. For this project, it is beneficial to design GANs which is less computationally expensive and produce data with high variance.

4. Approach and Implementation

This chapter is designed to discuss the underlying methodologies, creation of dataset, and architectural design for this thesis. As discussed earlier, GANs can be used to create new unseen datasets with varying inferences. It is viable to utilize this technology in the field of Agriculture which can benefit farmers in the improvement of crops yield and reduction of diseases. Many researchers are using the term "smart farming" which can take advantage of the internet of things (IoT) for real-time data extraction and analysis and machine learning for the classification tasks which can improve crop quality and reduce waste. There are numerous benefits of applications of machine learning in the field of agriculture. A few of these benefits are species recognition, species breeding, soil management, water management, yield prediction, crop quality, weed detection, and disease detection.

This chapter will explain the implementation of different variants of the GANs and also discuss their failures to produce high-quality synthetic images. The creation of the dataset will also be under discussion.

4.1 Creation of Dataset

Availability of the high-quality dataset plays an integral part in the training and evaluation of the machine learning algorithms. The success of any deep learning-based task lies in the better quality of available data. This means the data should not only be insufficient quantity for the coverage of the higher number of classes but it must have a good representation of the real scenario. Many companies provide different datasets for machine learning tasks. The quality of the datasets directly affects the quality of the GANs results. Some GANs require data in the paired form and some generate the data from noise. In all the cases, the quality and quantity of the dataset reflect the final output of the GAN.

In the field of agriculture, there is scarcity in the availability of the high-quality image-based dataset[Chiu 20]. There are many challenges in the collection of agriculture-based datasets. For example, the weather needs are good for the capture of images which means there must be appropriate sunlight, Cloudy overcast weather leads to low-quality images. After the advent of GANs, the generation of high-quality synthetic images has become less computationally expensive. It can save companies a lot of capital investment in the creation, management of datasets related to farming.

4.1.1 Field Image Acquisition

Image acquisition of outdoor farmland is a complex task to achieve because it requires a high-quality camera setup and ideal weather conditions. Different weather conditions have different effects on the images. In this thesis, it is important to make both domains semantically identical therefore GAN does not have to stretch comprehensively to map the source domain to the target domain.

For the collection of the image data, an apple orchard in Kallstadt, Germany was visited where researchers have used a special autonomous robot vehicle that can move between the fields with controlled speed and acceleration. There are two special stereo cameras mounted on the vehicle which records video feed at an output resolution of 3080×1080 with 30 fps. As it can be seen in fig 4.1 that these cameras are designed and developed by Stereo Labs which has depth-sensing technology. The total depth range is 0.4-25 meters. This allows capturing the images at an adequate distance from the apple tree because it is important to record maximum information to regenerate the images. These cameras play

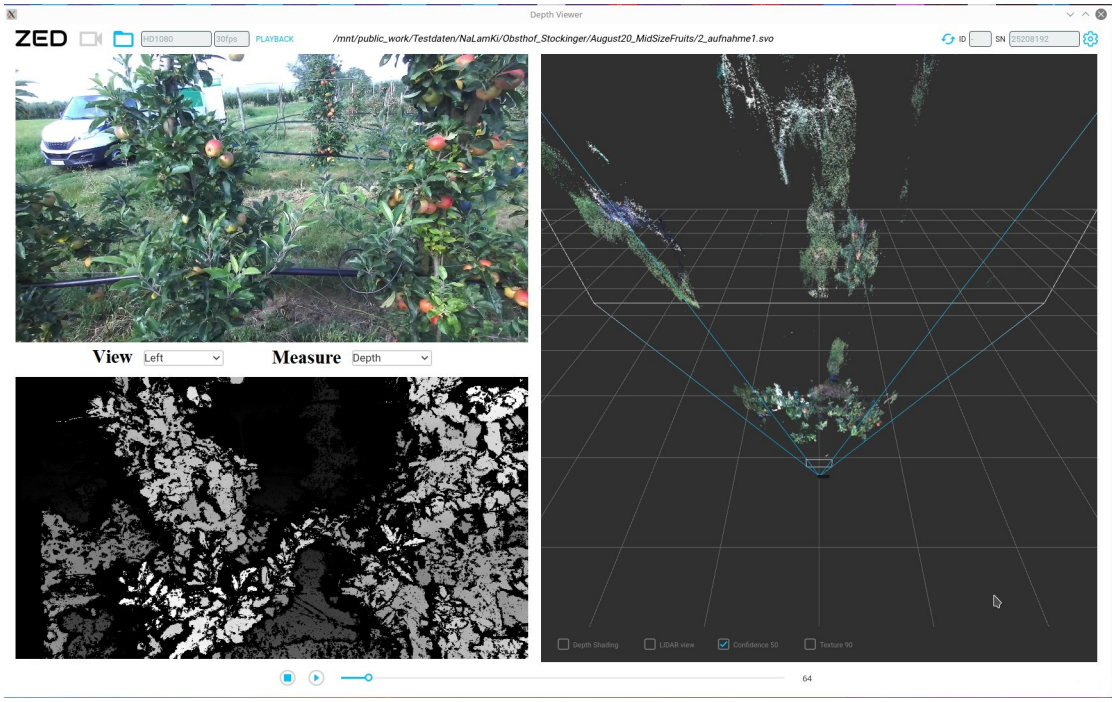


Figure 4.1: This is the screen grab of software development kit provided by stereo labs which allows to to capture the image with stereo cameras along with depth information.

a very important role in the identification of focused areas within the image. A camera mounted on an autonomous vehicle moves through the field, recording the video of apple trees with two angles. One stereo camera records the video at 60° and the other system records the video at about 10° from the ground. After getting the raw video feed with depth-sensing information, from the camera systems, a python based imaging pipeline is written for extraction of the relative images.

4.1.2 Imaging Pipeline for Data Extraction

As stated earlier a python based imaging pipeline has been written to extract the relevant data from the raw video feed. To achieve this, Python-based Open CV library [Bradski 00]

is used for the extraction of the images. OpenCV is an open-source library embedded with many image processing and computer vision-based algorithms to facilitate the researcher for a pre-processing pipeline. In this imaging pipeline, the software development kit provided by stereo Labs is used to read the frames from the .svo file format. The extracted frames are in RGB format which must be changed into binary images. in fig 4.2, it can be observed The binary image consists of only two colors which either black or white. The pixel value is either 0 or 1. After this step, morphological transformations are being

Figure 4.2: The RGB image is normal image which contains value of red, green and blue between 0-255. The Binary image contains only two values: 0 and 1 and gray-scale image only contains the intensity value of given pixel.

applied to the frames to remove any noise data. Morphological operations are performed on the region of the binary image which contains any shape. This operation requires two inputs: one is the binary image and the second is the kernel. To remove the nose, OpenCV provides two very critical operations: erosion and dilation. In the fig 4.3, the Erosion operation works by shrinking the boundaries of the foreground object. A kernel passes through the binary image looking for the pixel (0 or 1) and convert the pixel value to 1 only if all pixel under the kernel is 1 which makes the pixel close to the boundary of the foreground object to 0.

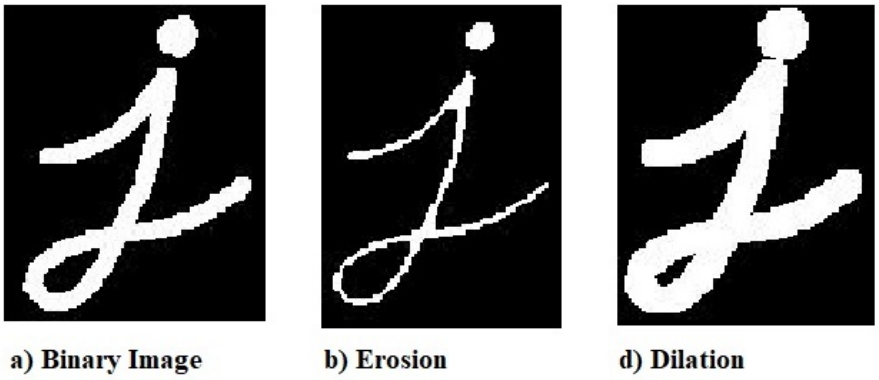


Figure 4.3: This visualisation explains the difference between erosion and dilation of binary image.

In the dilation operation, The mechanism is opposite to the erosion operation, which defines that pixel will convert to 1 if the only one-pixel value is 1 under the kernel. The effect of this operation is to increase the white region of the foreground object, which helps

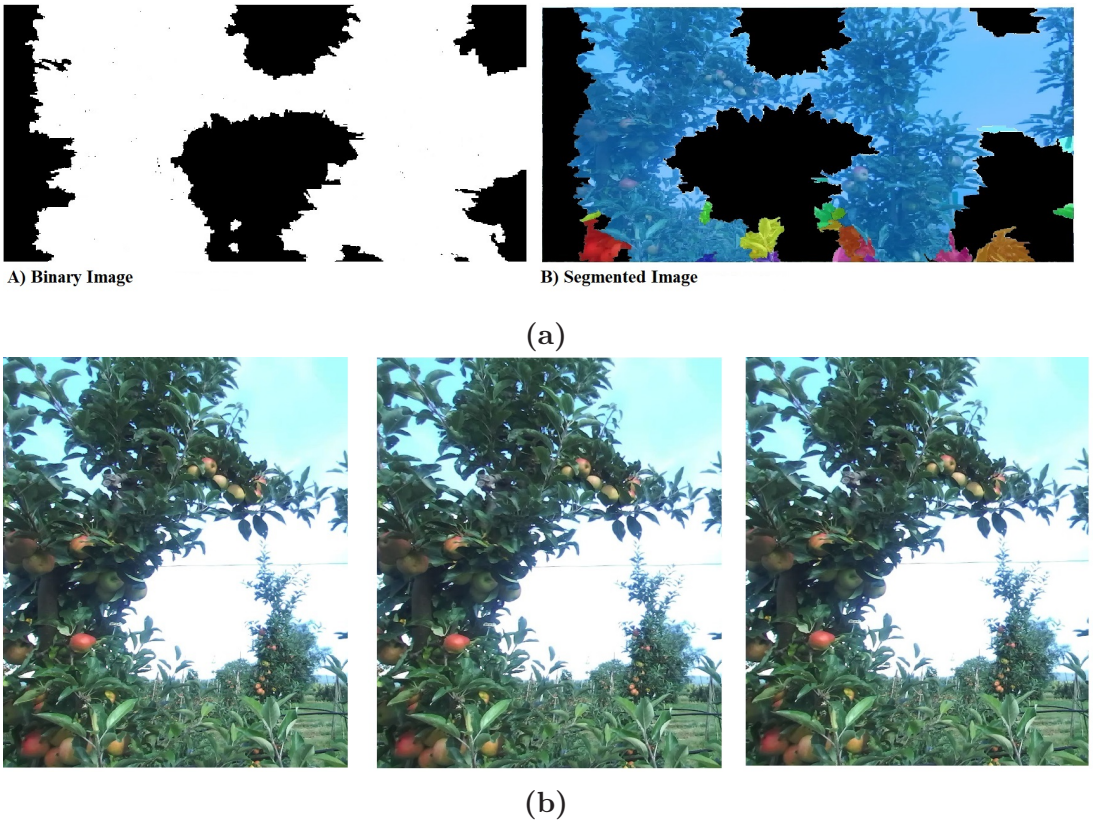


Figure 4.4: (a) Image shows comparison image of the application of watershed algorithm on apple orchard imagine data. (b) Image shows the final extracted images after the segmentation performed by watershed algorithm.

to remove holes within the object's region. The main objective of these morphological operations is to remove any noise from the binary image. This noise reduction allows to have better results in the segmentation as regions with the noise will not be segmented.

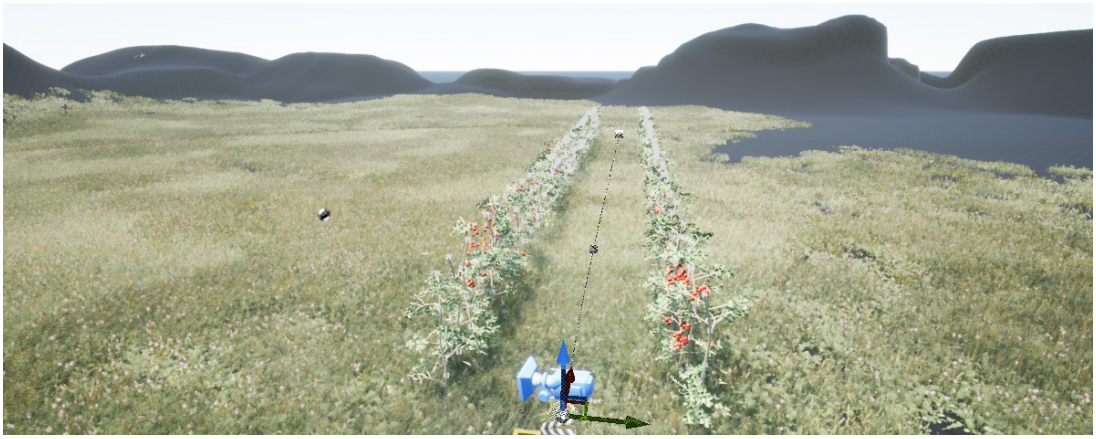
After the applications of these two morphological operations, the resultant binary image looks very sharp and contains no noise. This binary image works as a mask in the watershed algorithm which performs the segmentation on the RGB image based on the binary image. The watershed algorithm is a classical interactive image processing technique that utilizes region-based morphology. In this algorithm, a grayscale image is considered a topographical landscape that contains valleys and ridges. The OpenCV library includes marker based watershed algorithm which uses selected markers for the valley regions for the unification. It is important to select the foreground object in the image for the selection of the marker which will be considered as a valley. The selection of the foreground is done using depth-sensing information provided by cameras and morphological operations remove the noise from the binary image. The labeled foreground region and background region with 0. Apply the watershed algorithm to segment the portion of the image which contains apple trees. These apple tree regions are foreground images labels with 1 and background regions are the rest of the image which are labeled as 0.

The watershed algorithm allows the researchers to create segmented image-based data without the usage of deep networks. Segmentation can also be performed through neural nets however this is beyond the scope of the thesis as the main focus is the generation of

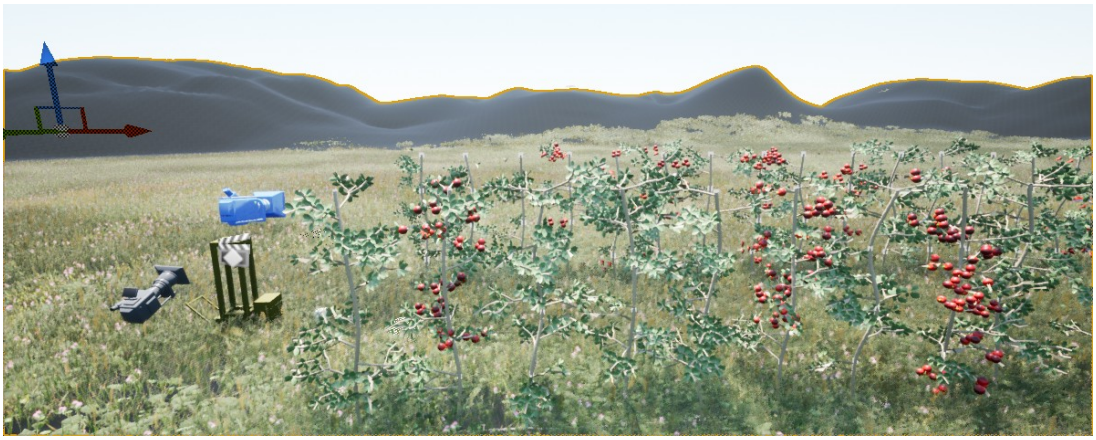
synthetic data through deep learning algorithms. It can be seen in the above fig 4.4a and 4.4b how the watershed algorithm helps to perform segmentation in the given apple tree images. This allows us to extract meaningful images from raw images of apple orchards. This pipeline is computationally inexpensive and less time-consuming in comparison to other deep learning algorithms for the segmentation. As it has been discussed that this thesis requires images of an only apple tree in a dataset so a GAN can take leverage from this information and perform the translation.

4.1.3 Creation of 3D Dataset

This has been established that under observation GAN architecture requires two datasets that belong to two domains. One domain can be addressed as a target domain that contains real segmented images of apple trees and the second domain can be addressed as a source domain that contains synthetic data designed in Unreal Engine [Sanders 16].



(a)



(b)

Figure 4.5: (a) Image is screen grab of unreal engine in which farm like environment is created. (b) Image is also the screen Grab of the unreal engine with different view angle.

In the fig 4.5a and 4.5b, it can be seen the 3D environment created in the unreal engine to acquire 3D images of apple trees. The Unreal Engine is a state-of-art 3D creation tool designed and developed by Epic Games. This tool has been helping researchers with the

creation of a synthetic environment for the entertainment industry for a very long time. It gives leverage to artists, enthusiasts to change any parameters of the 3D environment like weather conditions, textures, etc, without a physical inspection of the field. This tool is built on state-of-art technologies related to Computer Graphics and Animations. There are many 3D creation tools available in the market however unreal engine stands out amongst them because of high-quality community support and portability. With the minimum effort, Photo-realism can easily be achieved in 3D environments created through an unreal engine and can also be ported from desktop computers to mobile devices.

During the creation of 3D data, it was important to build and design a farm-like environment to achieve semantic similarity between real images and 3D images. This idea plays a very important role for the GAN to map the object from one domain to another domain while keeping the semantic nature of the domains intact. The first task is to create 3D models of Apple trees which have apples attached to them. The color of apple fruit can be changed from green to red and this thesis requires to transfer the style from one domain to another domain. for instance, the red color of 3D apple fruit must be mapped to the green apple of the real image using GAN. This would allow the creation of any kind of 3D data under any kind of environment. GAN needs a huge amount of data to generate the new synthetic data. Once the apple tree model is designed, 3D farm-like environment is created to place the models of an apple tree in the field. from fig, it can be observed that a 3D vehicle is created for the locomotion purpose to move between the fields of apple trees and the unreal engine's built-in camera tool is utilized for the acquisition of images. These 3d models of cameras are placed on the vehicle at a similar angle as at a real autonomous robot vehicle.

Figure 4.6: a) This is the image from unreal engine. b) This is the real image from apple orchard.

In the above fig 4.6, it can observe the final result of 3D image and real image of apple trees. Both of these images exhibit semantic similarity to achieve stability in mapping between two domains. It must be noticed that the color of the apple in a 3D image is dark red while the color of the real apple is green. This is where GAN will translate the style from a 3D image to a real image once the training is complete.

The light source in any environment has very significant effects on the image. Similarly, GANs can also generate faulty images if the light source is not at the correct position. In the unreal engine, there is three light position-controlled through the daytime plugin. In this plugin, it is possible to select the global location of the sun in the environment. It is a requirement that the main source of light must be behind the camera or right above the apple trees otherwise shadow effect can create artifacts in the generated image. To accomplish this task, Three daytime settings are selected in the unreal engine, morning, noon, and evening. In the morning, the main source of light is behind the camera and in noon it is right above the camera. In the evening, there is no direct light present in the scene but there is the presence of indirect light which creates a haze effect. The important statistics of datasets are given in below tabular form.

Dataset Apples Trees		
Daytime	3D Dataset	Real Dataset
Morning	10,000	8730
Noon	10,000	11,000
Evening	10,000	7,000

Table 4.1: This table represent the volume of training data. The variable figures of real data represent actual images extracted from imaging pipeline from raw video, captured in real apple orchard field.

The underlying architecture of GAN does not require to have a dataset in huge volume to get a feasible result. By this convention, 10k images were extracted from a 3D environment for the training purpose where a similar volume of images was extracted from a real environment of apple trees.

In the next phase of this thesis, different GAN architectures will be deployed and evaluated to achieve the desired results. If the desired results are not achieved then the reasons for failures are investigated through proper methodology. To achieve the results, progressive GANs were deployed which are designed to high-quality output with stability and variation. These GANs provide different training methodologies as compared to the normal training of GANs. In the next section, a deep discussion of Progressive GANs is followed for advantages and disadvantages.

4.2 Progressive Growing of GANs

The ultimate output of this thesis is to achieve the highest quality results with style transformed from the source domain to the target domain. To achieve this task, different GAN architectures are tested over the dataset created for these special tasks. Normally, Gans are complex systems that require carefully handcrafted methods to achieve the desired results. It is uncomplicated to recognize the Mona Lisa painting by Leonardo da Vinci than to redraw it without any error. Generative Models are relatively more complex because their task is to generate the image from latent noise and make it perfect iteratively while discriminative models are tasked to just classify the image. The training of the GANs is considered as a min-max game where one model is trying to maximize its performance and another model is trying to minimize the gains of the first model. The GANs exhibits

very instability towards parameter selection and also suffer from vanishing gradients when the discriminator gets too powerful during the training. In short, training of GANs is a very complex procedure and their stability is difficult to achieve due to their complex nature. If there is a requirement to generate the high-resolution image from GANs then it is difficult to achieve since the discriminator can easily classify between generated images and training images. This also significantly amplifies the vanishing gradient problem [Odena 17]. High-resolution generated images are trained through smaller batches. However, they also need high memory resources due to their size.

Researchers have attempted to solve this issue by publishing a new methodology for training. This is known as progressive GANs [Karras 17] and it is the first architecture that is deployed for testing. The main idea of this training methodology is to start the training with a minor generator and discriminator: start from the low resolution and then keep adding layers in the model as the training progresses as visualized in fig 4.7. This method increases the speed of training and protects the model from instability. When

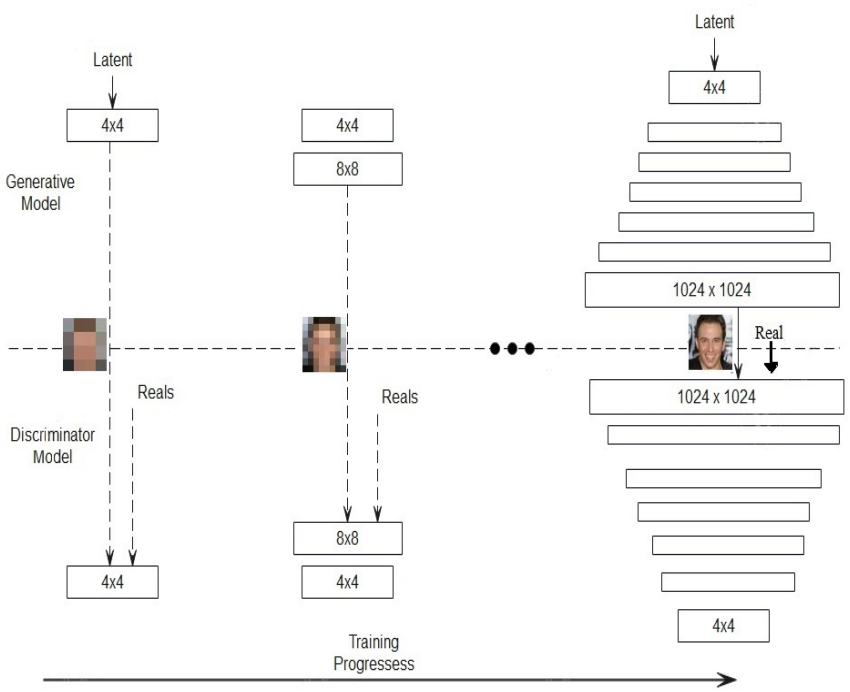


Figure 4.7: At starting point of the training, the generator (G) and discriminator (D) are producing the result in very low resolution of 4×4 pixels. As the training goes, further layers are added into the model G and D which result in the higher spatial resolution of generated images. On the right side, it can be observed with six generated examples at the resolution of 1024×1024

adding new layers, it is important to fade them in the model in such a way that it remains stable during the training. This incremental increase in the layers of the model avoids the sudden shock to the training of GAN. This incremental increase of layers allows the network to discover large-scale structures in an image during the initial phase of the training. After this step, the network moves to find the finer detail in the image. This allows the network to learn incrementally instead of learning it simultaneously. All

the layers in this methodology are trainable which allows the model to train from lower resolution to higher resolution.

This incremental addition in the network has multiple advantages. During the initial stage of training, smaller layers are more stable because there is less information about the underlying classes in the data. The network goes for finer details when the incremental increase of layers is done during the training thus stability of the network allows generating megapixel images at the resolution of 1024^2. Generation of higher resolution images also allow to use WGAN-GP [Gulrajani 17] or LSGAN loss [Han 16]. The training time is also reduced to 2-6 times depending on the target resolution.

4.2.1 Increase of Variation in Progressive GANs

Variation is an important part of the final result of any GAN. If the model can not generate different results each time then it means the model has not captured the variation in the image distribution. The ultimate goal of successful training of GAN is to produce different results each time and results must be within the training data distribution However GANs tend to capture a small amount of variation in the training data due to the complex nature of the architecture. Researchers have provided the solution which is known as mini-batch discrimination. This solution states that feature statistics should be computed across all the mini-batches of training images, in this way, the similarity between all the mini-batches will be shown. Implementation of this solution is done by deploying a mini-batch layer at the end of the discriminator. This solution requires no new hyperparameters or any other parameters. Furthermore, they calculate the standard deviation from every example across all the channels and all of the pixel values and then compute the mean value of standard deviation. Now it is feasible to replicate the mean value and concatenate it to all the spatial locations of the mini-batches which results in an additional feature map. This is a single channel that contains a feature map and concatenates this input data. Therefore, a single value is representing standard deviation across all the channels and all the pixels values.

It is not feasible to add a new layer directly to the model, during the training however It is important to add the layers without any replication effects. In the fig 4.8 (a) part illustrates transition of 16×16 images from generator(G) to discriminator(D). In the (b) part, the layers which deal with higher resolution images are treated like residual blocks whose weights "alpha" has a linear incremental range from 0 to 1. 2x and 0.5x means doubling the resolution of the image and cutting it half respectively. Upsample the 16×16 image using nearest neighbor and pass it through convolution layer which converts the channels into "toRGB". The other path is to take 16×16 image and pass it through 32×32 convolution layer. After these two paths, interpolation is computed by

$$result = (1 - alpha) \times Upsample + (alpha) \times Outsample \tag{4.1}$$

The new faded convolution layers takes more responsibility for the generation of the new image. The "toRGB" and "fromRGB" project feature vectors to the RGB plane and vice versa. To further explain this process, first introduce the 16×16

In the final generator part, the researchers fade in the new layer by down-sampling it with average pooling. After these procedures, researcher take 16×16 image upsample it 2 times and pass it through 32×32 "toRGB"

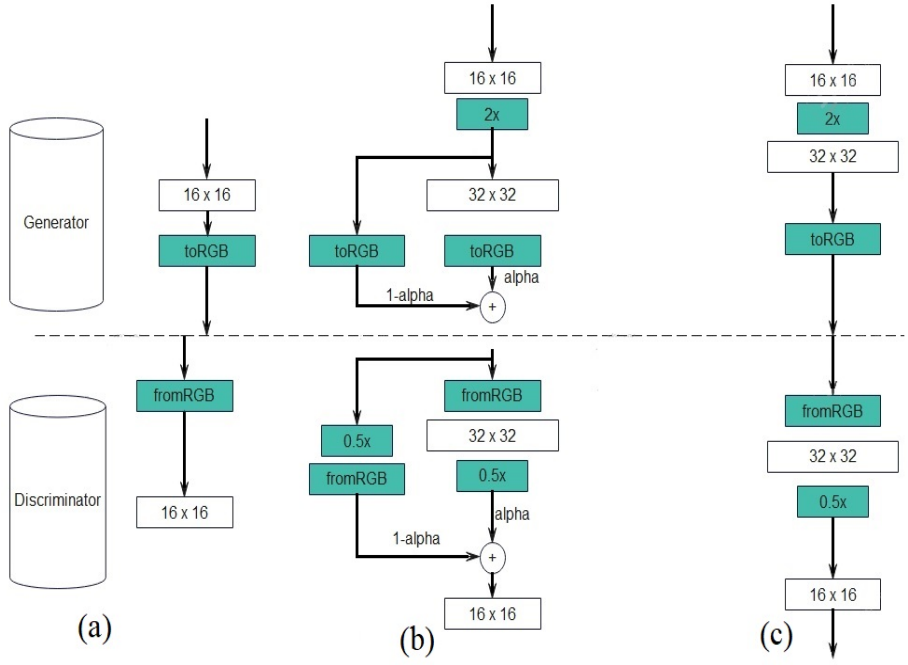


Figure 4.8: This visualization explain the how the layers are faded into the model. The transition from 16×16 (a) images to 32×32 (c) images happen in (b) where layers are treated like residual block and alpha is between 0 to 1. The fade in of layers are done through nearest neighbor up-sampling like $2\times$,$0.5\times$, in (b).

4.2.2 Normalization of Progressive GANs

It has been observed by the researchers that GANs have a habit of escalation of the signals. This happens due to unnecessary competition between two networks. During the training, researchers also observed this phenomenon which is called covariate shift. This is described as when training data and test data does not belong to the same distribution thus they have covariate shift in their standard deviation. To resolve this issue, a batch norm is introduced to normalize the input to each convolution layer. This provides a guarantee that distribution is similar as input to each convolution layer.

4.2.3 A Network Structure and Training Configuration for Apple Orchard Dataset

As stated earlier, the total number of training images available is around 10,000. This methodology does not require the dataset in pairs therefore real dataset of the apple orchard field is used for the training of progressive GANs. This architecture in fig 4.9 consists of two networks that have replicated 3- layers block that is introduced into the system one by one, during the training. The last convolution layer of 1×1 in generator block belongs to "toRGB" block in Fig 4.8 and first convolution layer of 1×1 belongs to "fromRGB". In this thesis, the training started with 4×4 resolution until 10,000 images have been shown to the networks. For the next iteration, the system introduces "fade in" in the first three layers and stabilizes the networks for 10,000 images, and then again "fade in" the next three layers. This process repeats itself until all the layers have not been visited. The latent vectors are randomly generated points on the 512-dimensional

hyper-sphere and represent the training images and generated images in the range of -1 to 1. In the beginning, the noise vector in the generator has $512 \times 1 \times 1$ z-dimensions and input channels are also equal to the 512. The next layer is called transposed convolution after the latent vector in the Generator. The size of this convolution layer is 4×4 which map the input to $512 \times 1 \times 1$. after the transposed layer, system introduces 3×3 which works on the 4×4. It must be kept in mind that this is the beginning of the training. For the activation layer, leaky ReLU is introduced with a leakiness value of 0.2 in all the

Generator	Act.	Output shape	Params
Latent vector	–	512 × 1 × 1	–
Conv 4 × 4	LReLU	512 × 4 × 4	4.2M
Conv 3 × 3	LReLU	512 × 4 × 4	2.4M
Upsample	–	512 × 8 × 8	–
Conv 3 × 3	LReLU	512 × 8 × 8	2.4M
Conv 3 × 3	LReLU	512 × 8 × 8	2.4M
Upsample	–	512 × 16 × 16	–
Conv 3 × 3	LReLU	512 × 16 × 16	2.4M
Conv 3 × 3	LReLU	512 × 16 × 16	2.4M
Upsample	–	512 × 32 × 32	–
Conv 3 × 3	LReLU	512 × 32 × 32	2.4M
Conv 3 × 3	LReLU	512 × 32 × 32	2.4M
Upsample	–	512 × 64 × 64	–
Conv 3 × 3	LReLU	256 × 64 × 64	1.2M
Conv 3 × 3	LReLU	256 × 64 × 64	590k
Upsample	–	256 × 128 × 128	–
Conv 3 × 3	LReLU	128 × 128 × 128	295k
Conv 3 × 3	LReLU	128 × 128 × 128	148k
Upsample	–	128 × 256 × 256	–
Conv 3 × 3	LReLU	64 × 256 × 256	74k
Conv 3 × 3	LReLU	64 × 256 × 256	37k
Upsample	–	64 × 512 × 512	–
Conv 3 × 3	LReLU	32 × 512 × 512	18k
Conv 3 × 3	LReLU	32 × 512 × 512	9.2k
Upsample	–	32 × 1024 × 1024	–
Conv 3 × 3	LReLU	16 × 1024 × 1024	4.6k
Conv 3 × 3	LReLU	16 × 1024 × 1024	2.3k
Conv 1 × 1	linear	3 × 1024 × 1024	51
Total trainable parameters			**23.1M**

Discriminator	Act.	Output shape	Params
Input image	–	3 × 1024 × 1024	–
Conv 1 × 1	LReLU	16 × 1024 × 1024	64
Conv 3 × 3	LReLU	16 × 1024 × 1024	2.3k
Conv 3 × 3	LReLU	32 × 1024 × 1024	4.6k
Downsample	–	32 × 512 × 512	–
Conv 3 × 3	LReLU	32 × 512 × 512	9.2k
Conv 3 × 3	LReLU	64 × 512 × 512	18k
Downsample	–	64 × 256 × 256	–
Conv 3 × 3	LReLU	64 × 256 × 256	37k
Conv 3 × 3	LReLU	128 × 256 × 256	74k
Downsample	–	128 × 128 × 128	–
Conv 3 × 3	LReLU	128 × 128 × 128	148k
Conv 3 × 3	LReLU	256 × 128 × 128	295k
Downsample	–	256 × 64 × 64	–
Conv 3 × 3	LReLU	256 × 64 × 64	590k
Conv 3 × 3	LReLU	512 × 64 × 64	1.2M
Downsample	–	512 × 32 × 32	–
Conv 3 × 3	LReLU	512 × 32 × 32	2.4M
Conv 3 × 3	LReLU	512 × 32 × 32	2.4M
Downsample	–	512 × 16 × 16	–
Conv 3 × 3	LReLU	512 × 16 × 16	2.4M
Conv 3 × 3	LReLU	512 × 16 × 16	2.4M
Downsample	–	512 × 8 × 8	–
Conv 3 × 3	LReLU	512 × 8 × 8	2.4M
Conv 3 × 3	LReLU	512 × 8 × 8	2.4M
Downsample	–	512 × 4 × 4	–
Minibatch stddev	–	513 × 4 × 4	–
Conv 3 × 3	LReLU	512 × 4 × 4	2.4M
Conv 4 × 4	LReLU	512 × 1 × 1	4.2M
Fully-connected	linear	1 × 1 × 1	513
Total trainable parameters			**23.1M**

Figure 4.9: This is the blueprint of the network design that is used for generator and discriminator for the generation of 1024×1024

layers of both networks, However, in the last layer, it is feasible to use linear activation functions. In this methodology, the researchers did not deploy either batch, layer, or weight normalization in both networks however pixel-wise normalization of the feature vector is deployed after each convolution layer 3×3 in the generator. It is important to set the bias parameters value to 0 and weights according to the normal distribution of feature data with unit variance.

For the training part, this model is trained using Adam optimizer with $\alpha = 0.001, \beta_1 = 0, \beta_2 = 0.99$ and $\epsilon = 10^{-8}$. It is not feasible to use any kind of learning rate decay. However, for visualization of the generator, at any given point in the training, it is advisable to use the exponential running average for the weights related to the generator with a decay value set to 0.999. For the resolution of $4^2 - 128^2$, mini-batch with the size 16 is utilized. In order to save available memory, it important to gradually decrease the size of mini-batch to

$256^2 \rightarrow 14,512^2 \rightarrow 6$ and $1024^2 \rightarrow 3$.The researchers of this work utilized 800,000 images on 4 VG100 GPUs for training purposes, 4 RTX 2080 Ti GPUs a are deployed with 10,000 images of real data. Due to the complexity of the training methodology, this architecture faces many instability issues. The images training at 1024 resolution needed more GPU memory which led to other problems. It was also observed that the mode collapse issue was raised during the training when at higher resolution the discriminator is over-fitted and classifying every image as fake, not allowing the generator to learn. For this reason, a new architecture was chosen for the thesis which is discussed briefly below.

4.3 Cycle-Consistent Adversarial Networks with image-to-image Translation

The broader perspective of this work is to facilitate autonomous vehicular machines to perform better machine learning-related tasks in the field of agriculture. However, For all the machine learning-related tasks, dataset plays a cerebral role, and the collection of agriculture-based data has many challenges. Teams must be sent on a field trip for the collection of data through imaging devices. It is fact that agricultural products are based on time series analysis which means a normal crop changes in size and shape at regular points in time. This change can be categorized from sprout stage to ripening stage. From the data collection perspective, it is difficult to collect the images of the crop at every stage and cycle of growth. To overcome this problem, it is feasible to use deep learning algorithms for the generation of the image data which comes from similar data distribution of crop.

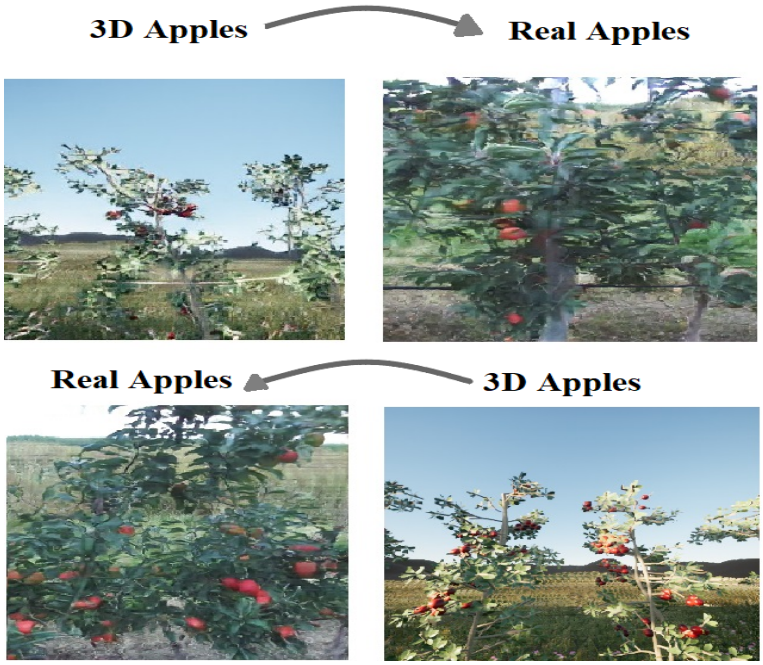


Figure 4.10: This is the visualization of mapping between unpaired data domains $X and Y$ computed through Cyclic GANs. It can also observed that 3D apple tree and real apple tree are semantically similar.

The main objective of this thesis is to generate images of apples with variance, from the real distribution of apple images collected from the apple orchard. These generated images are reconstructed through GANs which are unsupervised deep learning networks designed to replicate the training data distribution. There are many versions of GANs available however this thesis requires a special version like image-to-image where images from one domain can be mapped to a different domain, as can be seen in fig 4.10. In this fig, it can be observed that mapping between zebra to the horse is easily computed through cyclic Gans. The domain zebra and domain horse are semantically similar. The Image-to-Image translation belongs to the class of vision and computer graphics problems where the objective is to learn the mapping from one domain to another domain. This domain can also be semantically dissimilar which means the network does not require data in the paired form. This is especially helpful in those cases where the training data is not available in the paired form. To accommodate this problem, researchers from Berkeley AI Research Lab have presented an approach [Zhu 17] in which mapping is learned between the image from source domain (X) to images from target domain (Y). The researchers the main objective is to learn the mapping function or translation function ($G : X \rightarrow Y$) using adversarial loss, in such a way that image data points from $G(X)$ are indistinguishable from data distribution (Y). Due to the under-constrained nature of this translation, the researchers provided a solution of inverse-mapping ($F : Y \rightarrow X$) and also presented cyclic consistent loss for the enforcement of $F(G(X) \approx X$.

As it can be seen in fig 4.11, data is categorized between three portions: one portion is called paired data in which images from the target domain and images from the source domain have an absolute semantic similarity. For example, on the left, it can be observed that the target domain x is the edge feature map of the source domain y. They are semantically indistinguishable which means the mapping function will have minimum difficulty to transfer the style from target to source domain. On the other hand, paired data has no semantic similarity in which the target domain can belong to any class and the source domain can belong to a different class. This architecture is designed to transfer the style from unpaired target domain X to source domain Y. To achieve this task, the new cyclic loss is introduced to maintain the quality of generated images. For this thesis, a semi-unpaired term is coined in which both target and source domain exhibits approximate semantic similarity. This means the source domain has images from the unreal engine where the apple orchard field is designed in a controlled environment while the target domain has real images from the real apple orchard. It is computationally easier for the Cyclic GANs to transfer the style from the target domain to the source domain. In the field of agriculture, it is very difficult and expensive to obtain paired or even semi paired images of the crop, for the GANs. To overcome this problem in this thesis, 3D models of apple trees are placed in a simulated environment, provided by an unreal engine, and then images are obtained from internal camera systems for the creation of a synthetic dataset.

4.3.1 Pix2Pix Network in Cyclic GANs

Due to the unsupervised learning nature of cyclic GAN, the model needs to learn the mapping without any assistance from source and target data distribution. For this reason, the researcher developed forward and backward cyclic consistency which states that when the translation happens between the first domain to the second domain then the reverse mapping should get spatially similar or semantically similar images from the first domain.

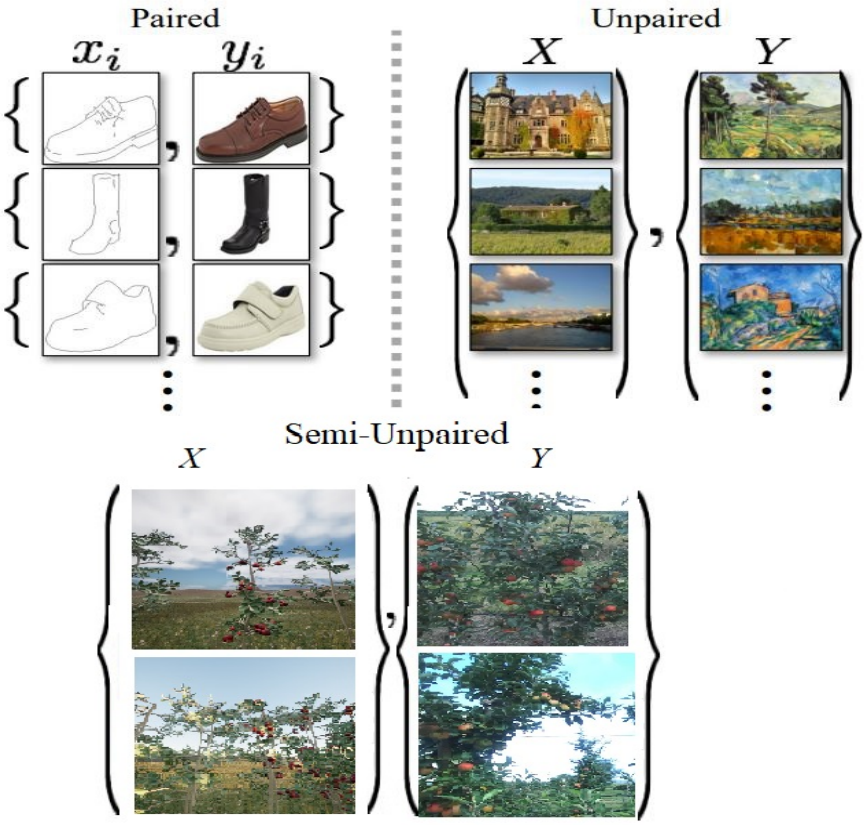


Figure 4.11: As it can be seen on the left that Paired training data are semantically similar while Unpaired data (right) has no semantic similarity between them. Semi-Unpaired (below) data has one domain from unreal engine and another domain from real images. Both of these domain represents apple orchard field.

This consistency allows keeping the check on the cyclic model for not deviating from objective.

The intuition behind this model comes from the "pix2pix" framework from researchers [Isola 17]. This pix2pix model comes from condition adversarial GANs which states that there must be at least one condition under which GANs can learn mapping. Generator model is convolution neural network with U-net model which is used for generation of images and in discriminator model, patch-GAN is used. The choice of patch for discriminator model describes that only patches will be penalized for non-consistency instead of the whole image.

In the fig 4.13, it can be observed that pix2pix architecture looks similar to general GANs architecture. It is important to take an example of the black and white picture to understand the network's architecture. As it can be seen in the fig 4.13 that input black and white image is processed by generator model which results in the production of the color image as final output. In the generator, input black and White image go through different convolutions filters along with up-sampling operations which converts the colorless image into a 3-channel color image. After the generation of a color image, the synthetic image is concatenated with the original input black and white image which converts the

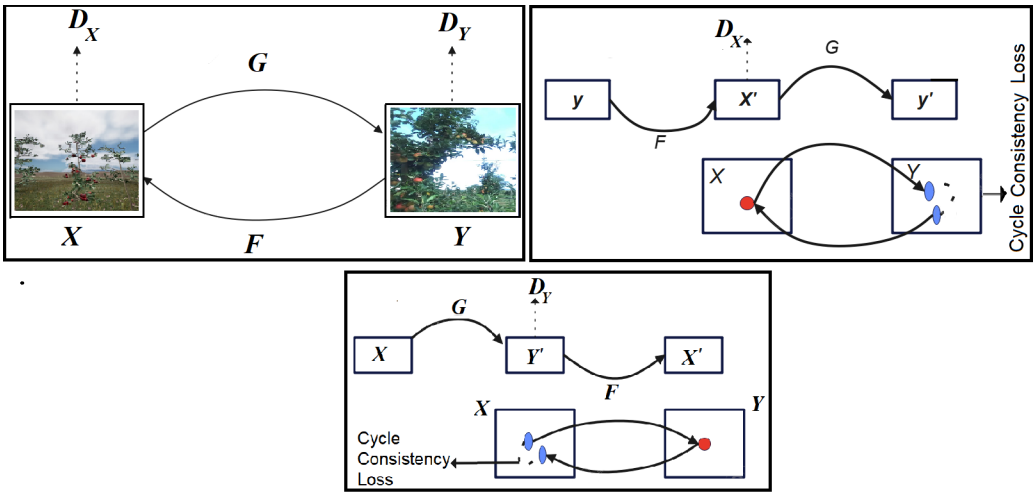


Figure 4.12: (a) In Cyclic model, there are two mapping functions ($G : X \rightarrow Y$) and ($F : Y \rightarrow X$) along with respective adversarial discriminators D_X and D_Y. This setting, D_X motivates generator G to translate X domain image to output image which is similar to domain Y images. This is also true for D_X and F. Due to unpaired datasets, researchers have introduced two cyclic consistency loses which provides cyclic consistency between target and source domain. (b) this is called forward cyclic consistency loss $X \rightarrow G(X) \rightarrow F(G(X)) \approx X$ and (C) is called backward consistency loss which is written as $Y \rightarrow F(Y) \rightarrow G(F(Y)) \approx Y$

image with 4 channels. This concatenated image tensor is given to the discriminator network which is of type patch-GAN which produces the output image with size $N \times N$.

The patch-GAN discriminator gives $N \times N$ predictions. These predictions are many overlapping patches in a source image. For example, for each 70×70 patch of the input image, 30×30 is the output image that is given to the loss function. This loss function then compares this 30×30 tensor with a zero matrix. This procedure is referred to as generational loss. Both black and white image and their replica color image is used for the calculation of real loss from the given dataset. This real loss is a sigmoid cross-entropy of $N \times N$ output from the generator.

To understand the pix2pix architecture which is the basis of Cyclic Gans, it is important to understand the U-net architecture because it is being used by the generator. The design and development of the U-net [Zhou 18] happened because of the problems related to image segmentation. In semantic segmentation, every pixel of the image belongs to the given class. The goal of this segmentation is to designate the important portion of the image to some class. This helps in classification tasks. The main use case of this network is in biomedical sciences where medical imaging segmentation is required for the classification tasks. U-net has the ability to focus on the patch of image for segmentation. It can expertly localize the region and also distinguish the region. In the fig 4.14, it can be observed that the left part of the u-net is considered to be an encoder that is designed to encode the input image into feature maps at a different level. This is achieved through some convolutions functions and max pooling. The decoder block is designed to protect the lower resolution feature map onto pixel space. To achieve this task, the feature maps pass through the up-sampling and concatenation.

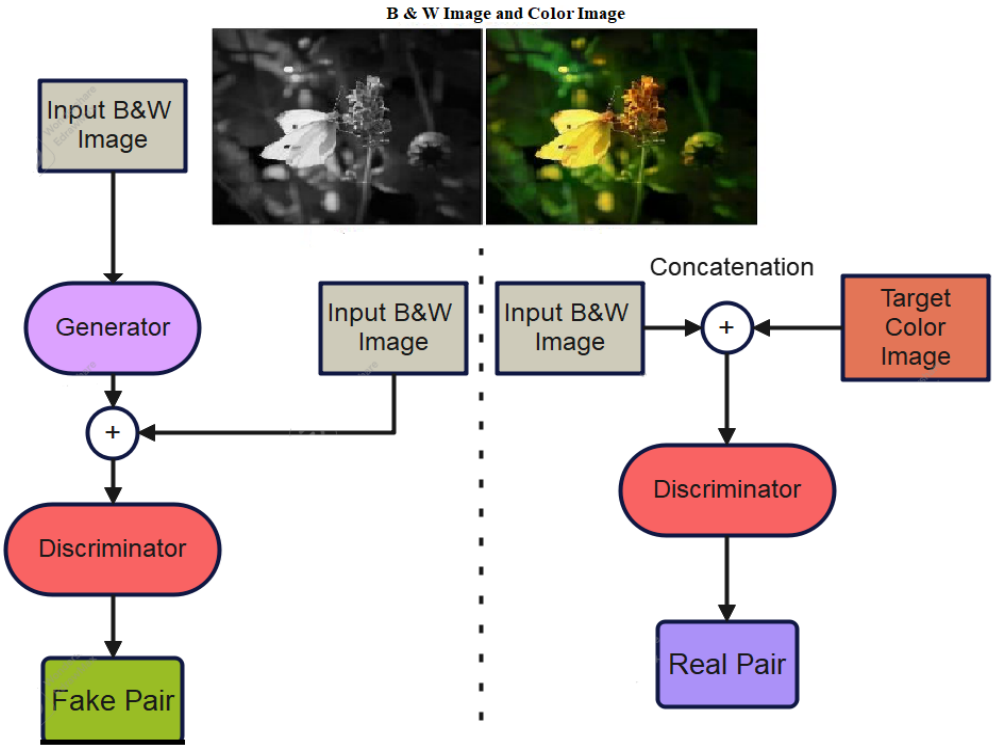


Figure 4.13: This is the visualization of pix2pix architecture with translation example of black and white image to Color image

4.3.2 Formulation of Cyclic GANs

To understand the pix2pix architecture which is the basis of Cyclic Gans, it is important to understand the U-net architecture because it is being used by the generator. The design and development of the U-net [Zhou 18] happened because of the problems related to image segmentation. In semantic segmentation, every pixel of the image belongs to the given class. The goal of this segmentation is to designate the important portion of the image to some class. This helps in classification tasks. The main use case of this network is in biomedical sciences where medical imaging segmentation is required for the classification tasks. U-net can focus on the patch of image for segmentation. It can expertly localize the region and also distinguish the region. In the fig 4.14, it can be observed that the left part of the u-net is considered to be an encoder that is designed to encode the input image into feature maps at a different level. This is achieved through some convolutions functions and max pooling. The decoder block is designed to protect the lower resolution feature map onto pixel space. To achieve this task, the feature maps pass through the up-sampling and concatenation.

$$Loss_{GENERATOR}(G, D_Y, X, Y) = E_{y \sim p_{data}(y)}[logD_Y(y)] + E_{x \sim p_{data}(x)}[log(1 - D_Y(y)G(x))] \tag{4.2}$$

$$Loss_{GENERATOR_r}(F, D_X, Y, X) = E_{x \sim p_{data}(x)}[logD_X(x)] + E_{y \sim p_{data}(y)}[log(1 - D_x(x)G(y))] \tag{4.3}$$

In the above equation, the generator G attempts to generate images $G(x)$ which are indistinguishable from to the images distribution from Domain$Y$. The discriminator D_y

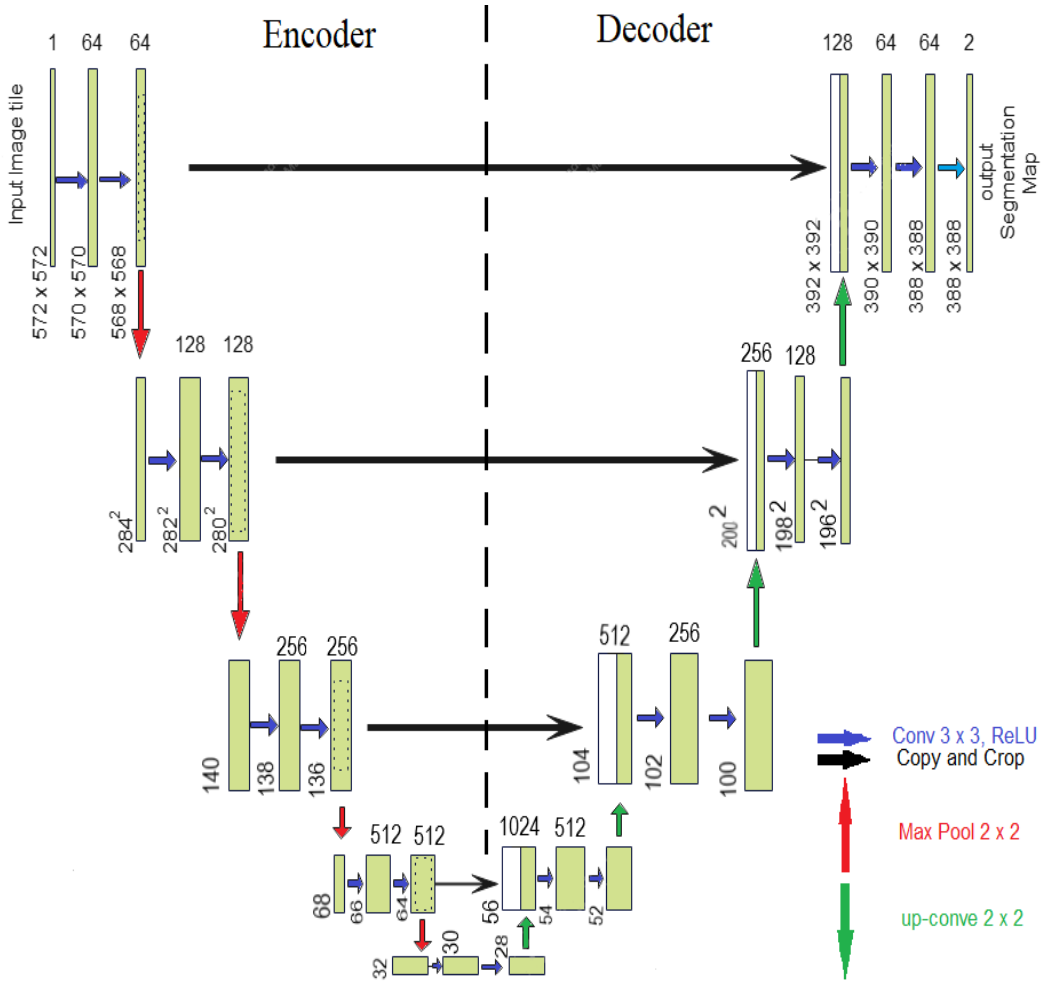


Figure 4.14: This visualization shoes the complete U-net architecture where yellow boxes are designated as feature maps and number of channels are written at the top of the boxes. The resolution of the image shoes at the lower left edge of the box. Different operations are denoted by the arrows at lower left of the image

attempts to classify between translated image samples $G(x)$ and real image distribution y. The Loss $Loss_{GENERATOR_r}(F, D_X, Y, X)$ is the similar loss function but from domain$Y$ to domain$X$. The min-max game of this GAN is defined as G attempts to minimize the objective loss while D tries to maximize it. This analogy of min-max game can be can be written as

$$min_G, max_D = Loss_{GENERATOR}(G, D_Y, X, Y) \tag{4.4}$$

and for loss function from domain y to domain x

$$min_F, max_D = Loss_{GENERATOR_r}(F, D_X, Y, X) \tag{4.5}$$

4.3.3 Cyclic Consistency Loss in Cyclic GANs

As shown above,a training with adversarial loss can learn translation G and F that generates image data points which similarly distributed as target domain X and$Y$ Cycle consistency loss is defined as when network generate images from source domain to target

domain then it must generate back from target to source domain. The loss value from domain$X$ to domain Y must be approximately equal to the loss value from domain Y to domain X. However this cycle consistency requires G and F to act like stochastic functions. These networks have large space which allow them to map the input images to any random permutation of images in target domain distribution. There fore adversarial loss functions are not enough to make sure that cyclic consistency remains intact which means there is no guarantee that learned translation function will map the input x_i to desired output y_i. To resolve this issue of randomness of mapping functions then these function should by cycle consistent. To make the formulation from this theory, it can be written as $x_i \rightarrow G(x_i) \rightarrow F(G(x_i) \approx x_i$ which is referred as forward cyclic consistency. For backward cyclic consistency, it can be written as $y_i \rightarrow G(y_i) \rightarrow F(G(y_i) \approx y_i$. It is important to increase the intensity of this loss using well defined cyclic consistency loss.

$$Loss_{cyc}(G, F) = E_{x \sim p_{data}(x)}[\|F(G(x) - x\|_1] + E_{y \sim p_{data}(y)}[\|F(G(y) - y\|_1] \tag{4.6}$$

The above equation is formal representation of the cyclic consistency loss function which restrict the mapping space of the translation function G and F. After the implementation of cyclic loss, it can be observed that generated image $F(G(x)$ is indistinguishable from input image y.

4.3.4 Compete Objective Fiction of Cyclic GANs

The complete objective function can be written as :

$$Loss_{complete}(G, F, D_X, D_Y) = Loss_{GENERATOR}(G, D_Y, X, Y) + \tag{4.7}$$

$$Loss_{GENERATOR_r}(F, D_X, Y, X) + \lambda Loss_{cyc}(G, F)$$

The λ variable in the above equation controls two objective function which can be written like this:

$$G^*, F^* = arg \quad min_{G,F} \quad max_{D_z, D_Y} Loss(G, F, D_X, D_Y) \tag{4.8}$$

4.3.5 Neural Style Transfer in Cyclic GANs

The core functionality of the image translation is to adapt the artistic style present in source domain to target domain without introduction of any artifacts. There are many approaches available in which style transfer function learns the style mapping between two images. In the Cyclic GANs, it is not feasible to learn the style mapping between only two images rather between both domain's images. For the neural style transfer, a prominent work by researchers[Johnson 16] from Stanford university is adapted in this thesis. This work has shown impressive results in neural style transfers and super resolution. The architecture of this network contain three convolution blocks and many residual blocks. To turn the feature map into RGB, using above fig 4.15, two and half $\frac{1}{2}$ strided convolution and one full convolution filter is used. Since the targeted output from the cyclic GANs is 256×256, it is feasible to use 9 blocks.For the implementation purpose, instance normalization is utilized. For the patch-GAN which is used as discriminator, it in important to take 70×70 patch for the GANs. This allows the discriminator to classify between real or fake patch of 70×70. This setting allows to transfer the style from the target distribution to the source

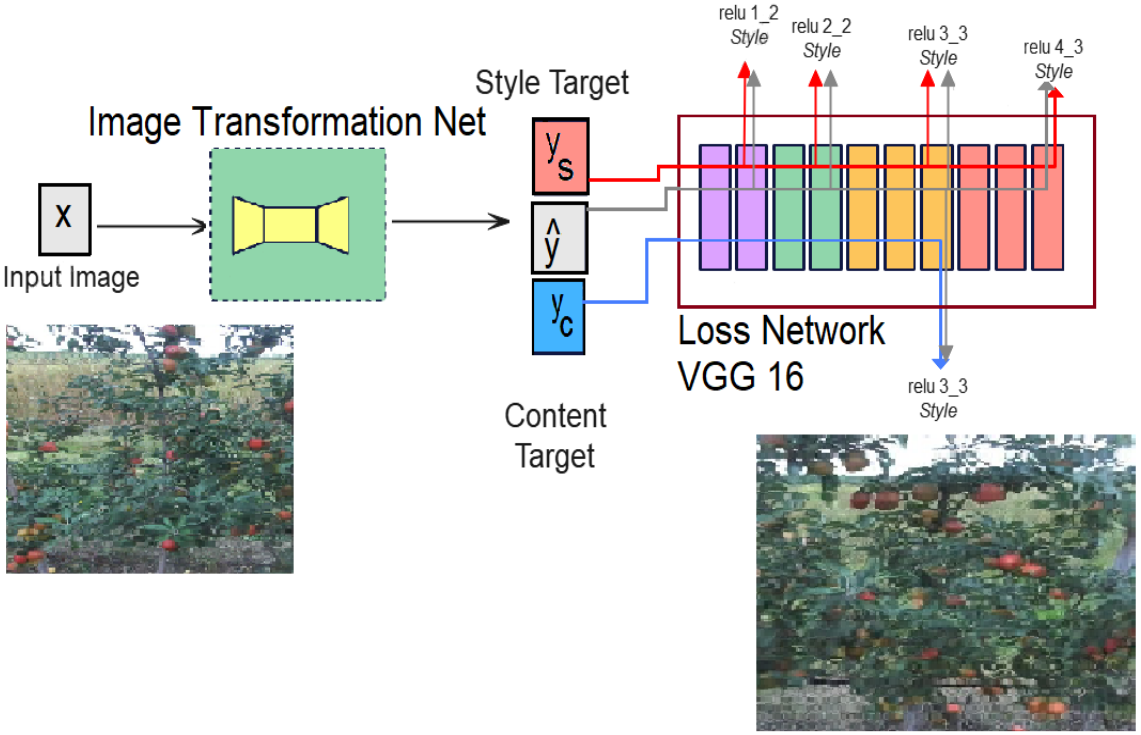


Figure 4.15: This the Visualization of VGG loss Network along with image transform Net. This loss function allows the image classification for the perceptual loss functions and it remains fix during the training function.

distribution as a collective which is very important for this thesis. First cyclic GANs learn the mapping between target distribution and source distribution then it must also perform the style transfer correctly in order to reconstruct the data which is indistinguishable from real data distribution.

4.3.6 Implementation of Cyclic GANs

As discussed before that architecture of the cyclic GANs is borrowed from "pix2pix" architecture, in which pared dataset is used for the training procedure. The pix2pix model belongs to class of image-to-image problems, which is designed to learn the mapping of an images that are spatially mirror to each other. The cyclic GANs are designed to learn the mapping between domains which are spatially and semantically distinguishable. The implemented cycle GANs contains similar blueprint of pix-to-pix architecture

For this thesis, similar architecture design has been borrowed from "pix2pix". In the fig 4.16, the generator part is sub divided into encoder, residual block and decoder. Basically, Generator and discriminator must start from the zero state, during the training. In the beginning, start with normal convolution 64 filters with 7×7 size and two down-sampling filters with 128 filters,3×3,stride 2 and 256 filters, with size of 3×3, stride 2 respectively. This down-sampling steps help in reduction of spatial information of the feature map which further helps in the less computation of less parameters.This reduction also helps to retain the information. After this step, introduction of residual blocks take place which allow the traveling of information from initial layers to final layers. The intuition behind

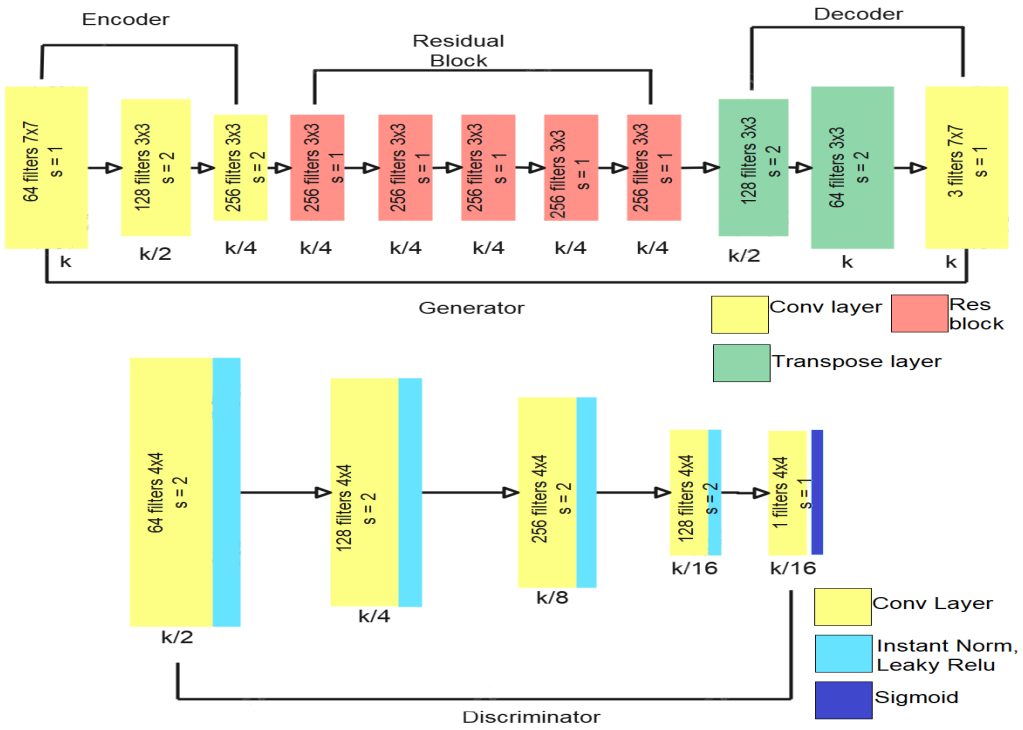


Figure 4.16: This is visualization of implemented architecture which is divided into generator and discriminator

the res blocks is to protect model from facing degradation problem. These res blocks also help to skip the shallow layers and transfer the spatial information directly to the end layers. There 6 res blocks implemented in this model which contain 256 filters of size 3×3 with stride of 2. After the res block, the introduction of up-sampling takes place using the transpose convolutions layer. In this layer, there are 128 filters with resolution of 3×3 and stride $= 2$. In the last step, a normal convolution layer is added for the mapping of current layer to the RGB channel.This whole generator is sub divided into encoder and decoder. In the encode part, convolution layer and down sampling layers allow to encode the spatial information information. In the decoder portion, similar information is recovered through up-sampling layers. For the discriminator part, there are 4 convolution layers, all with the stride of 2. These layers are known as Patch-GANs because the values are not singular value rather 3×3 grid. In this grid, each value corresponds patch in the original RGB image. For the generator loss function in this cyclic model, there are two generators models and two discriminator. In the generator loss, there standard generator adversarial loss when the models learns the mapping between 3D apple and real apple. The researchers of this model are using least squares loss. After the first loss, there is cyclic consistency loss which convert the image from 2D apple to Real image and then convert back to 3D apple. The other two losses are just the reverse of the above loss. The python based Pytorch library was used for the implementation of this model. The code is subdivided into separate classes for generator, discriminator, dataset and train.

In this thesis, the cyclic model is designed to learn the mapping from images of 3D apples to real images. The model should also be able to learn the style, colour space transfer from these two domains. Researchers designed this model to work with paired,unpaired or

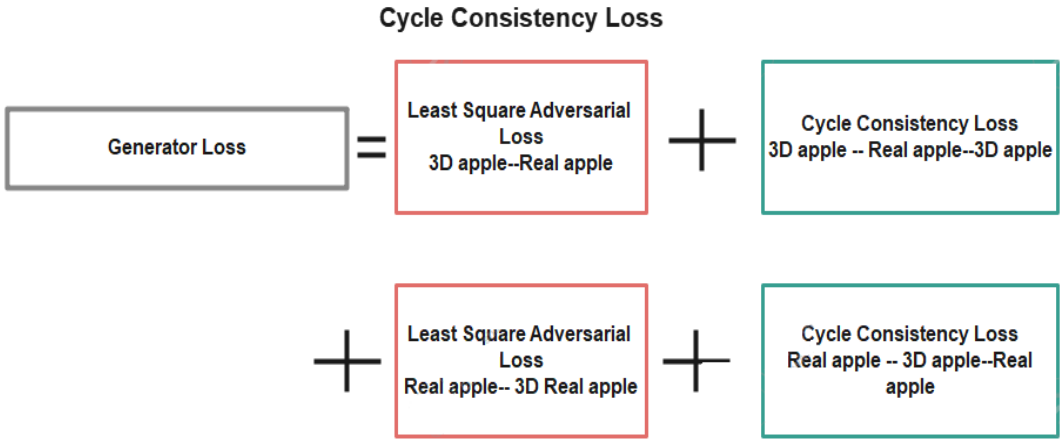


Figure 4.17: This is the visualization of loss function with cyclic consistency for 3D apple,Real apples

semi-paired data. However, in image-to-image translation, GANs are prone to produce unwanted result because they are learning the mapping from two domains. By looking at the results from other researchers, it was important to design the dataset which is semantically equivalent to the real data set. In this way, there will be no unwanted artifacts learned by the mapping function in the final generated images. To design the 3D apple orchard dataset in unreal engine, the geometrical structure of the 3D apples remains equivalent to real apples. The daytime and weather in the simulated unreal engine's environment is also similar to daytime and weather of real apple orchard. The underlying assumption is that 3D images must be semantically similar real apple image. However, if the colors of apple are changes from red to green then this information must be mapped between the domains and generated image must have green and red apple attached to the tree.

4.3.7 Training For Cyclic GANs

Training of the cyclic GANs contains many challenging tasks which include two mapping functions and three loss functions. It is very important to keep the model stable during the training since it can face many challenges like mood collapse or varnishing gradient issues. A normal GAN is prone to be unstable during the training due to the min-max game-like situation where one network is trying to acquire gain while the other network is trying to minimize the gain. Both of these networks must remain in an equilibrium state to converge.

As stated earlier, the cyclic model contains two generators and two discriminators which are trained simultaneously during the training process.It can lead to very complex situations to deal with. Researchers have necessarily installed cyclic loss function for the check on model stability. In the fig 4.18, there are two generators which are referred as G and F. The generator G takes the input from image data distribution that belongs to 3D apple orchard designed in unreal engine. The convolution network G takes this input and convert into real apple orchard image. The generator F takes the generated image from G and convert it back to original image which 3D apple orchard. The discriminator D_y performs classification that weather the generated image belongs to real apple orchard

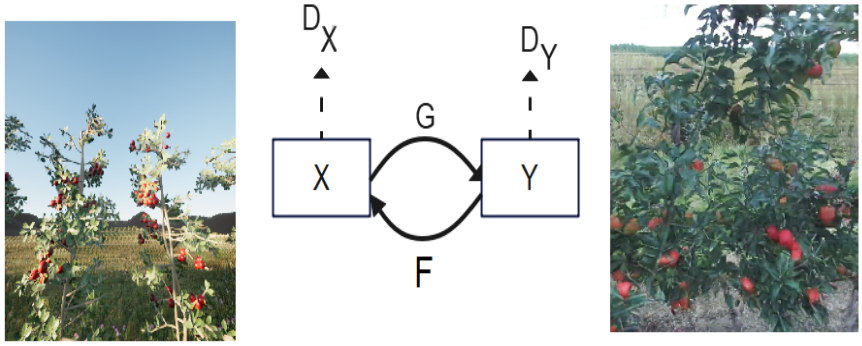


Figure 4.18: This is the pictorial representation of the Cyclic GANs from abstraction level where G and F are two generators and D_X andD_Y are discriminator

image distribution or not. The discriminator D_X checks weather the generated image belongs to image distribution of 3D apple orchard or not. This is standard setup of GAN except there are double the amount of generators and discriminators.

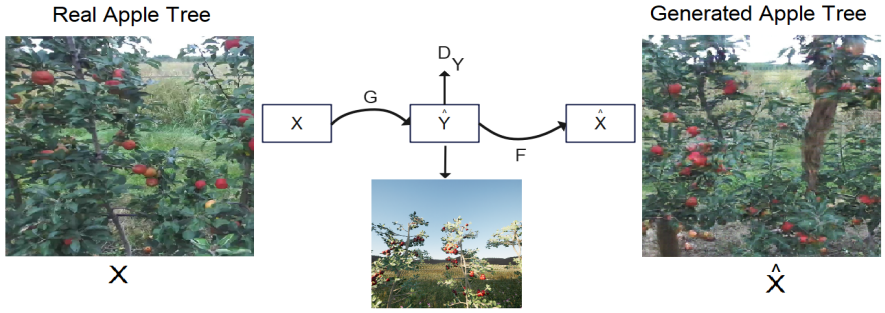


Figure 4.19: This is the detailed visualization of preservation of cyclic consistency

In this model, there is another loss function apart from discriminator loss functions, which ensures the better quality results from generator. In the fig 4.19, it can be observed how the consistency is maintained using this cyclic consistency check. In this setting, if generator takes the input from the real imaging data distribution then it is necessary to get the similar output after the one step generation. The one step generation in this fig, is defined as when generator G takes the real input of apple orchard's image and convert into 3D apple orchard output. The generator F takes output of generator G and convert it back to input of generator G which is image data from real image data distribution. The cyclic consistency is very important step in training of cyclic GANs. This step makes sure the cyclic model is heading towards right path. It can be observed in 4.20 that the cyclic consistency loss is the image differences between generated image and real image. It must be keep in mind that there is no proper way to visualize the difference between two images if there are semantically similar. A GAN has the built-in property that it must generate the data with variance which means every data point must be spatially different from previously generated image however, semantically it must be equivalent. During the training procedure, some features of the image comparisons are observed numerically in order to compute the loss within the model. This loss helps to drive the model towards

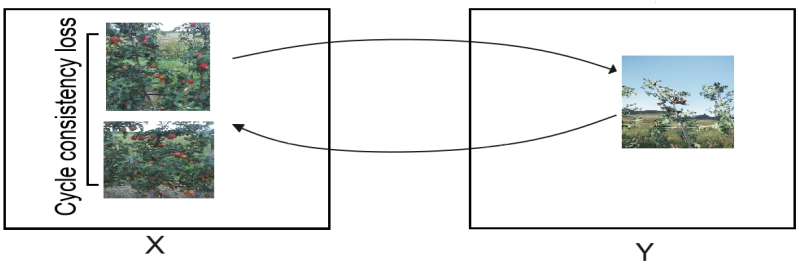


Figure 4.20: This visualisation represents the cyclic consistency when the apple image is generated from generator is compared with original apple image

convergence however this is not the same convergence that normally observed in CNN based model.

In the course of the training of cyclic GANs, the loss function $Loss_{GENERATOR}(G, D_Y, X, Y)$ for generator is used to minimize the $E_{x \sim p_{data}(x)}[(D(G(x)) - 1])^2$ and discriminator is used for maximization of the function which is referred as the $E_{y \sim p_{data}(y)}[(D(y) - 1])^2 + E_{x \sim p_{data}(x)} D(G(x))^2$. To control the oscillation, it is important to discriminate through the history of generated images instead of the latest generations of images. For this thesis, the total amount of images (3D images + real images) is close to 20k. Images that are designed by the unreal engine, are always equal to the real images from the apple orchard. As discussed earlier, an unreal engine allows changing all variables in its simulated environment which gives complete control over the image generations. There are three light sources set in the simulated environment through an unreal engine: one is behind the camera which means morning daytime is simulated. When the light is behind the camera, the images are crisper and their spatial dimensions are more visible to the network. In the second set, the light source is directed at the 90° which allows some shadows in the picture. In the third set, the light is behind the apple orchard field which gives the evening effect. This is done to introduce variance in the dataset which is good for the network. The value of $\lambda = 10$ is used with a batch size of 1. Adam solver is deployed because of its efficiency in computational resource management. The training of this model ran for 200 epochs to get proper convergence. The total training time for cyclic GANs on the custom-made dataset is 7 days with the RTX 2080Ti GPUs.

In the fig 4.21, it can be observed that discriminator's loss graph oscillated around 0 but it must be noted that GANs do not have convergence as compared to other deep learning methods.In GANs, there is no loss function as the loss function is itself a neural network which is being trained along side generator. This is the reason why the authors of cyclic GANs did not provide any training graph. To get better idea of the training procedure, it is recommended to output the generated images at equal intervals of times which allows proper visual inception by human subject. If the generated images at equal intervals are similar then this means the model is facing mode collapse issue. If the quality of the generated images are being deteriorate, during the training then this means model is not going towards convergence. It must be noted that this convergence is not like regular convergence of conventional deep learning methods.

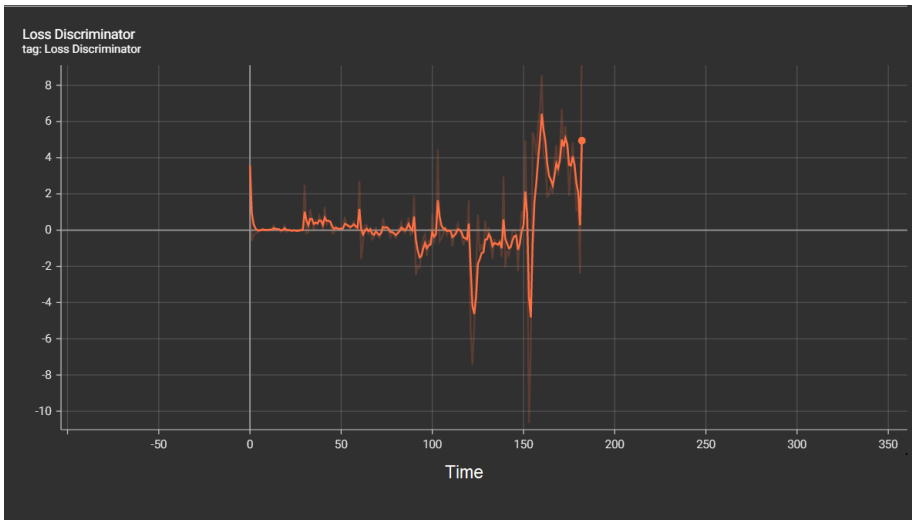


Figure 4.21: This is the graph of discriminator's loss in cyclic GANs, during the training. It can be observed that cyclic GANs loss network usually oscillates around zero except in few occasions which is considered normal behaviour in the training, however, discriminator's loss graph is not true depiction of GANs.

4.3.8 Fréchet Inception Distance for Custom Synthetic Data

Fréchet Inception Distance (FID) [Heusel 17] score is the metric system that is designed to assess the quality of the generated images from the GANs. This metric system is used for the computation of the distance between calculated feature vectors for real and generated images. The FID score defines the similarity index between real and generated images. This score informs about the quality of the images. A low score indicates a higher similarity between two data distributions. The FID score shows similarity over the computer vision features of the raw images by using the inception v3 model. This v3 model is used for image classification.

FID is working extension of inception score (IS) [Salimans 16]. The researchers of this paper wanted to eliminate the subjective human evaluation which means the group of people is used for the physical inception of the GANs output images. For this issue, they [Szegedy 16] purposed the use of a pre-trained deep neural network model trained on Image-Net for the image classification. The image classification procedure is specifically designed for the generated images from GAN. In this inception V3 model, the probability of class belongingness for each image is calculated and predicted. These predictions are used for the final inception score. This inception score can be divided into two categories. One is known as Image quality in which the score tells whether the images look like some specific object or not. In the category, image diversity is checked which states how many ranges of objects are generated in the image. This v3 model is designed to compare the conditional label distribution with the marginal distribution. The mathematical formulation of the FID is given below:

$$IS = exp(E_{x \sim p_g} D_{KL}(p(y|x)||p(y))) \tag{4.9}$$

A generator in the GAN must exhibit two qualities to deceive the discriminator: One it must generate images with semantic objects so that conditional label distribution $p(y|x)$ is low entropy and second, it must generate diverse images so marginal distribution

$\int_x p(y|x)p_g(x)$. In the case of IS, a high score represents better quality while a low score represents that the generator is not able to reconstruct feasible outputs. The intuition

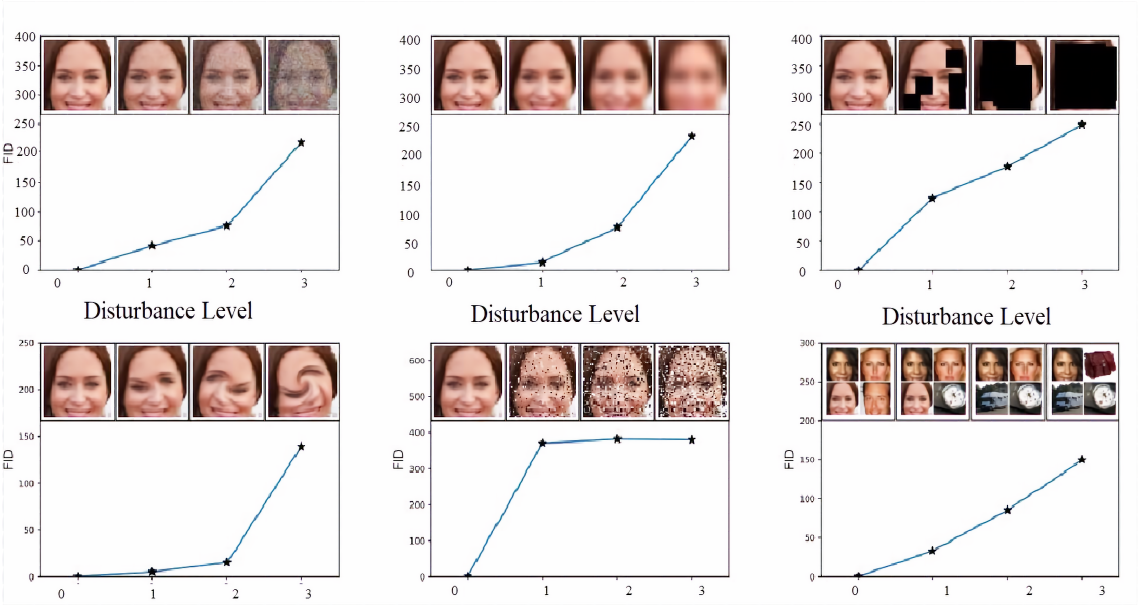


Figure 4.22: This is the visual comparison of FID scores when the image is being iteratively degraded.

behind the inception score is to estimate the quality of the collection of synthetic images. This estimation measure is based on the classification score by the V3 model and the classified object must be one of the 1000 known objects trained in the V3 model. The inception score is the combination of conditional class predictions and integral of the marginal probability. There is a limitation with the inception score which leads to the development of the FID score. In the inception score, the statistics of real-world image distribution are not compared with statistics of generated image distribution. It is believed that to get a good quality measure, statistics from the real-world data must be compared with the generated data for the final score. The mathematical formulation of this concepts is given as:

$$FID = ||\mu_r - \mu_g||^2 + Tr(\sum_r + \sum_g - 2(\sum_r \sum_g)^{1/2}) \tag{4.10}$$

where real data $X_r \sigma N(\mu_r, \sum_r)$ and generated data $X_g \sigma N(\mu_g, \sum_g)$. Real and generated data distributions has total 2048-dimensional activation of the inception v3 model thus lower FID score is better since which is calculated by distance between the activation distributions. As it can be seen in fig 4.22, the lowest FID score corresponds to better high-quality images. For the evaluation purpose, the FID score is state-of-art in the field of unsupervised generative deep learning models. Every research paper related to GANs is using FID to assess the quality of its generated output. Many critics argue that the FID score is not the true definition of quality assessment for generated images however, there are no globally agreed performance metrics present so this metric is used.

4.3.9 Use Cases of Cyclic GANs

Cyclic GANs are designed to learn the mapping between two domains which are semantically different. This allows multiple use cases in the field of computer graphics. The researchers

have used different datasets in order test the usability of the model where the paired dataset is available. Cyclic Gans are designed to work on paired and unpaired dataset but their true strength lies int the translation between unpaired data. They also have the ability to learn the input data distribution along with style mapping between target and source. In the next subsection , performance of the cyclic GANs will be briefly discussed and visualised on paired and unpaired data.

4.3.10 Style Transfer with Paired Dataset

The paired dataset gives the model supervised learning since both domains are semantically similar. Both domains exhibit similar probability distributions which allow the cyclic loss to be minimum and the model has enough space to reproduce the same results. For the paired data , CMP facade Dataset [Tyleček 13] is used which contain labels-to-photos. For another paired data, UT Zappos50k dataset [Yu 14] is utilized for the edge-to-shoes mapping because it contains shoe images along with respective edge images. As it can be

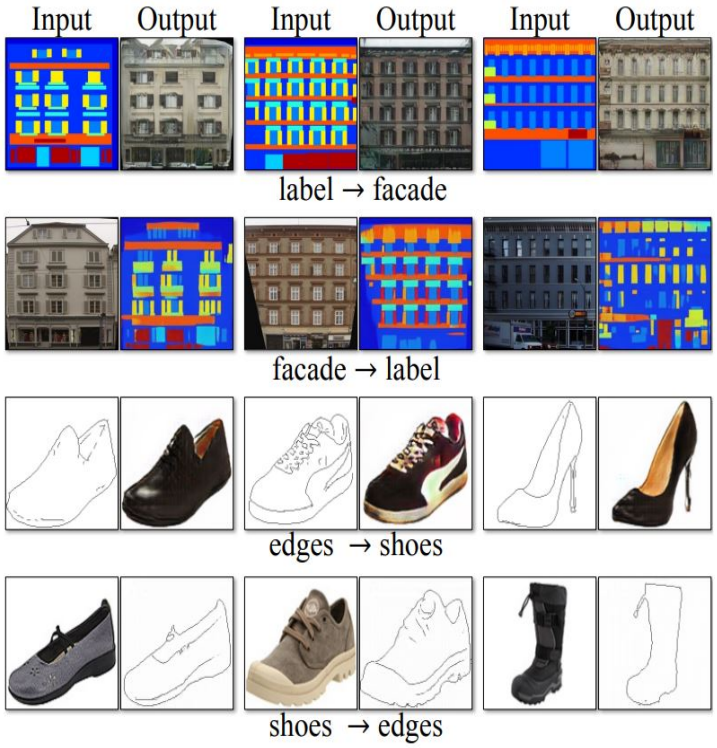


Figure 4.23: This image shows the results of cyclic Gans over paired dataset and it outperforms "pix2pix" model in certain cases.

seen in fig 4.23 that paired dataset is used for the training purpose to learn the mapping between semantically similar domains. The results show very good output images.

4.3.11 Style Transfer with Unpaired Data

When the cyclic GAN is trained on landscape imagine data which comes from WikiArt and Fliker, it produces the result by mimicking the art style from the entire collection of

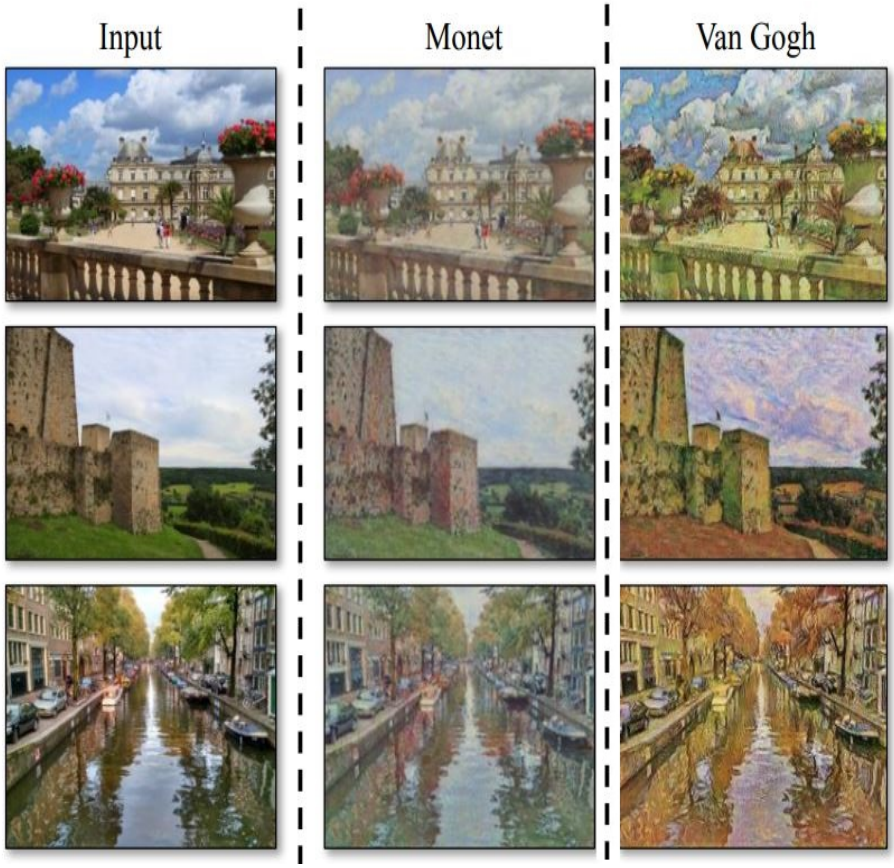


Figure 4.24: It can be observed in this image that Cyclic GAN with unpaired dataset, has successfully converted the landscape images into art painting from Van Gogh or Monet.

artworks. In this setting, source input belongs to real landscapes photos while the target input comes from renowned painters like Van Gogh.

It can be observed from the 4.24 that any image from any domain can be converted into any art piece. This is also know as style transfer between two unrelated domains.The Monet and Van Gough are famous painters from the history and their painting style is different from each other. However, cyclic GANs have made it possible to transfer the style vectors from paintings of these painters to the landscape images. In this thesis, style transfer is done by mapping the style vector from 3D images of apple orchard to real images of apple orchard. To achieve this task, semi paired data set is constructed for the training process. In which, a cyclic GANs will learn the style vectors of 3D apples images and transfer these vectors to newly generated images.

4.3.12 Object Transfiguration with Cyclic GANs

Object transfiguration means an alteration in the shape or appearance of any object through a change in molecular structure. Many existing deep learning-based object transfiguration methods show reasonable performances however they are specially designed for this task only. The cyclic GANs provide object transfiguration as one of the use cases without the requirement of paired data configuration. To achieve good results in object

transfiguration, datasets of two different classes, from Image-Net[Deng 09] are used for the training purpose.

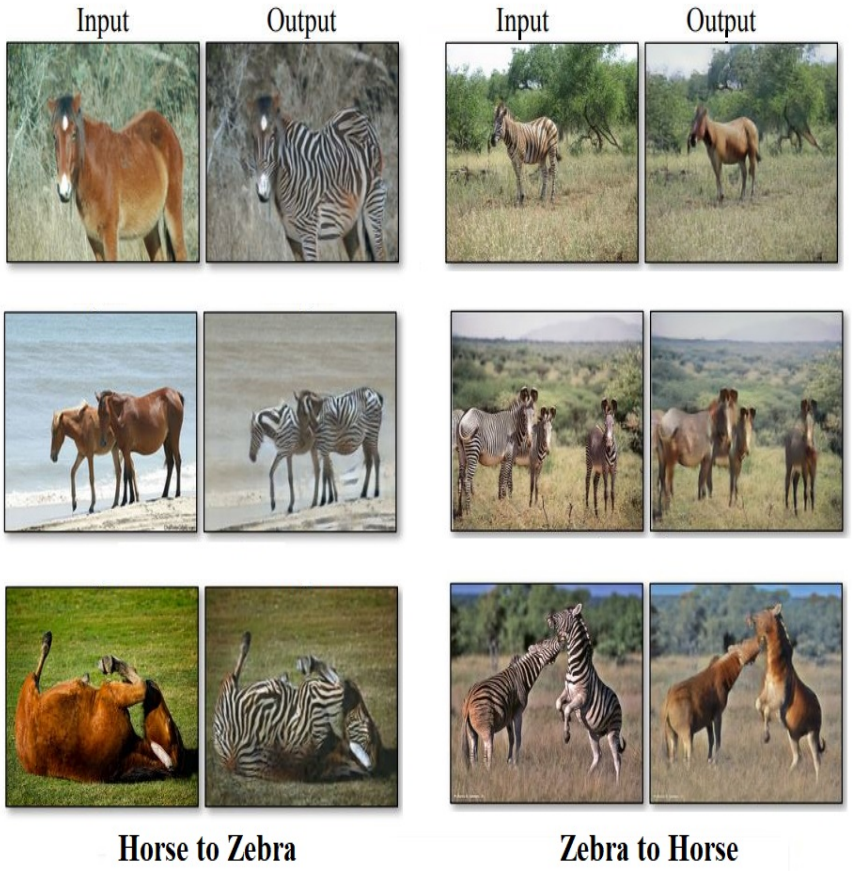


Figure 4.25: In the image, Object transfiguration is achieved through Cyclic GANs where texture from zebra is translated to horse object and vice versa.

It can be observed in the image 4.25 that cyclic GANs have achieved very good results in the object transfiguration when both classes are semantically similar. It is important to test this model on classes that are not semantically similar.

In fig 4.26,it can be observed that when the classes are not semantically, the cyclic GANs produce marginal or average results. Object transfiguration is a complex task to achieve in the computer graphics field. Deep learning Generative models have provided different solutions however there is a proper evaluation metric available that can determine the quality of the translated images. When the classes are semantically similar e.g, orange and apple are semantically similar. They are different in color however their shapes are more or less indistinguishable and it is less complex for the mapping function to learn the internal structure of the domains. Therefore, it can be seen that results are very good between horse and zebra in fig 4.25 since both domains are indistinguishable in shape. When the cyclic GANs are exposed to semantically distinguishable classes then it is very complex for the model to incorporate the spatial mapping from target to source distribution. Researchers have tried object transfiguration between different domain and their results are not very good. To address this issue, researchers [Turmukhambetov 15]

purposed subspace model which is designed to learn the mapping of internal features of the objects from both domains while keeping the external features the same.

Figure 4.26: In this image, researchers have tried to learn the mapping from Ramon to Human face through cyclic GANs.

In this thesis, Object transfiguration also plays a very critical role since cyclic GANs must translate the structure of 3D apples designed in an unreal engine, to real apples from an apple orchard field. However, in this custom dataset, domains are semantically indistinguishable which means each domain contains apple trees with red apples on them so it is easier for the model to train the mapping function between these domains. Images of apple trees contain complex structures like leaves, stems, and apple fruit, therefore it is still difficult for GANs to regenerate the complex structure by learning from the input data distribution. In this model, cyclic consistency loss guides the generation with complex structures like trees with leaves and stems. This allows the model to maintain the structural integrity of the object within the image however sometimes it produces results that can be regarded as faulty images because of the probabilistic nature of the GANs. In nutshell, GANs are designed to learn the probability distribution of the channels in the imagine dataset and they must reproduce the data which can lie in the area under the curve of probability. This means sometimes data produced by GANs can be distinguishable from the source or target domain. In the following fig 4.27, it can be observed that cyclic GANs produced fairly good results of mapping between apple and oranges. Both of the domains are semantically similar which allows the model to extract maximum performance

in the generation of the image. After further inspection, It is observable that complex structures like the apple tree, leaves, and stem have been maintained during the training. These example images are best selected among the generated data therefore, to maintain

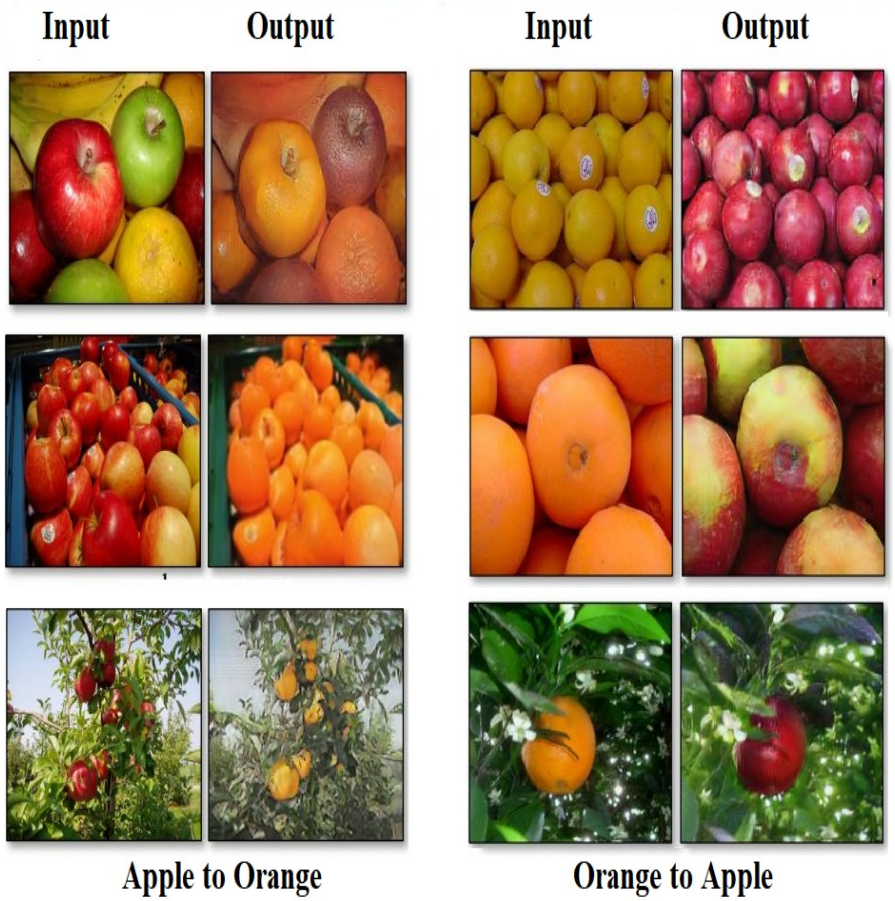


Figure 4.27: This image shows the result of the learned mapping function between apple to orange and orange to apple

the variance in the cyclic model, it can be expected from the model to generate images that can not be related to either apple or orange.

4.3.13 Photo Enhancement Feature in Cyclic Model

In this thesis, the output resolution of the images generated through cyclic GANs is 256×256, which means the model has to reproduce the result within the same resolution which can affect the quality of the images. It has been observed that this model has produced some artifacts within the images of the apple trees. There can be two ways to address this problem: one is to alternate the architecture design however this is beyond the scope of this thesis, another way is to introduce higher resolution images 600×600 in the target domain which will allow the model to learn the mapping from 256×256 to 600×600 of the apple images dataset. In this way, images generated from the model will have higher resolution and enhanced colors.

Cyclic Gans can be used for the enhancement of the images by utilizing the higher resolution images as the target source. Many quality-related photo variables can be achieved through

Figure 4.28: As it can be seen in this image that photo enhancement is achieved by introduction of higher resolution of similar images as target data distribution.

this model. As it can be seen in the fig 4.28 the Depth of field effect is introduced in the generated images. This is achieved by taking similar images from a professional digital single-lens reflex (DSLR) camera and using it as target distribution in the training procedure of cyclic GANs.

5. Experiments and Results

In this chapter, the evaluation of GANs is discussed in length. GANs are very effective deep learning systems that are used for generating data in an unsupervised manner. GANs are consists of neural networks which work against each other to get better. The generator is a convolution neural network that is designed to generate images to deceive another convolution neural network in the system. The other convolution neural network is known as a discriminator which has a relatively easier task of classification between real image and generated image. Both of these networks are trained to gather from zero to convergence. As it can be seen that there is no objective loss function that can be deployed to train the generator network, instead discriminator is used as a loss function to assess the quality of the generator network.

There are many metric systems available that can be used for evaluation purposes. These metric systems can be subdivided into quantitative and qualitative metric systems. It is important for this thesis to have a qualitative metric system because the quality of the generated images of an apple orchard is important rather than the quantity. During the process of this thesis, there were two models tested on the custom apple orchard dataset. One Model is Progressive GANs and the second model is Cyclic GANs. Progressive GANs have their limitation and are very unstable during the training. The failures of this model will be discussed briefly in this chapter. Cyclic GANs produce very good results which will also be evaluated in this chapter.

5.1 Limitation of Progressive GANs

In this thesis, the basic requirement was to use such variants of the GANs which produce the best results with stable training, high variance, and minimum computation requirement. However, higher resolution image generation can introduce many problems during the training of the model. The discriminator can easily classify between real and generated images due to high resolution. This can limit the growth of the generator's training and can make the discriminator stronger. This leads to the problem of mood collapse. GANs are very complex deep learning models which are hard to train. They consist of two networks that are involved in the min-max game during the training. This complexity leads to many

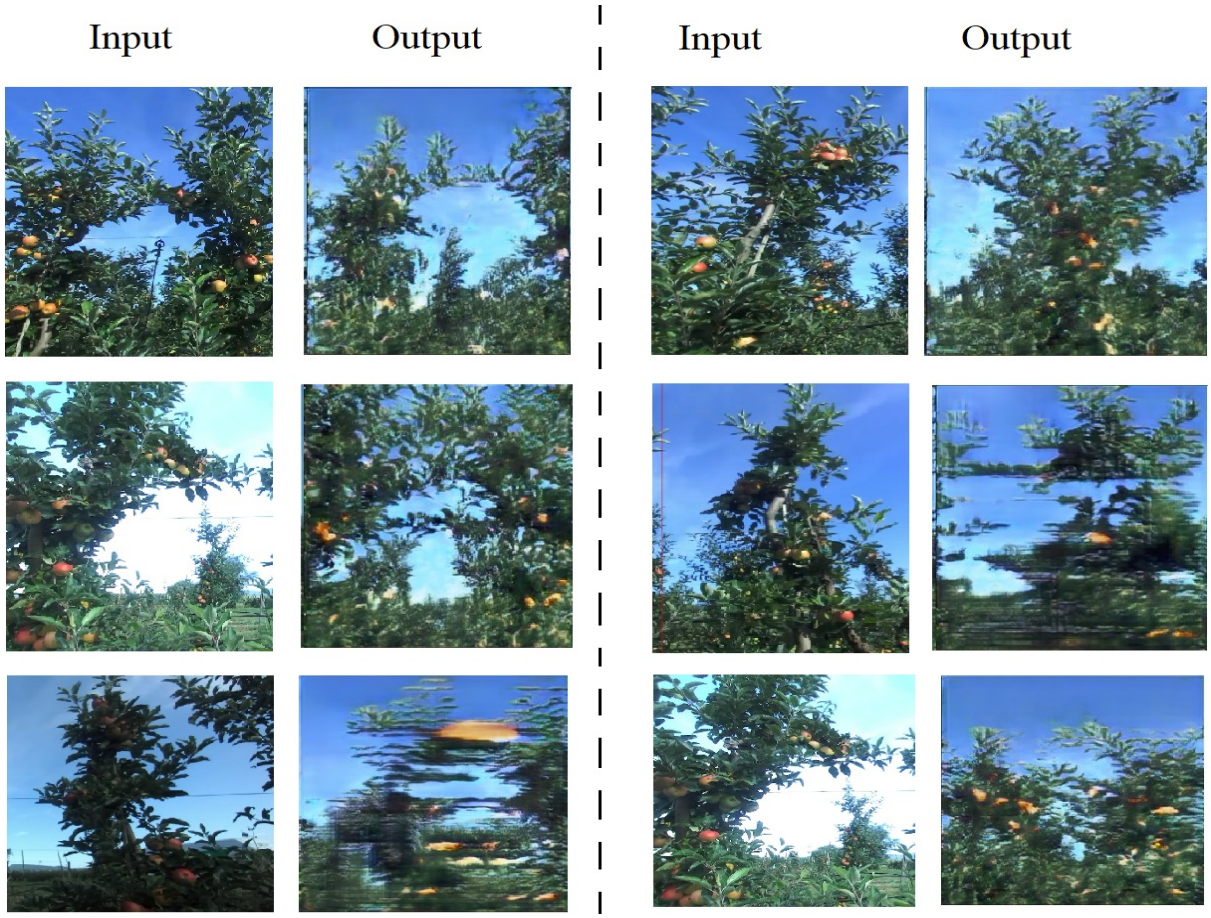


Figure 5.1: This is the visual comparison between real data and Pro GANs generated data.it can be observed that there are many geometrical artifacts in the resulted images which overfits the discriminator's learning.

FID Score of Pro GANs	
Datasets	FID scores
Apple Orchard Dataset 1	310.3435435678
Apple Orchard Dataset 2	390.3453453445
Apple Orchard Dataset 3	335.546710924

Table 5.1: This table shows the FID scores of Pro GANs where higher scores represent lower quality of the image generated. During training when Pro GANs move from low resolution images to higher resolution images, discriminator overfits the classifications of generated image which hampers the learning of generator.

problems. Progressive Gans are designed to produce the generated results with higher resolution and with stable training. They are not special GANs variant with intriguing architecture designs rather, they are training methodology in which both generator and discriminator starts from the lowest resolution, and later on, new layers are added into the network to capture internal details of training images. They are also not image-to-image translators which makes them infeasible for this thesis. The researchers of this work has used massive computational power along with a huge volume of the dataset as compared

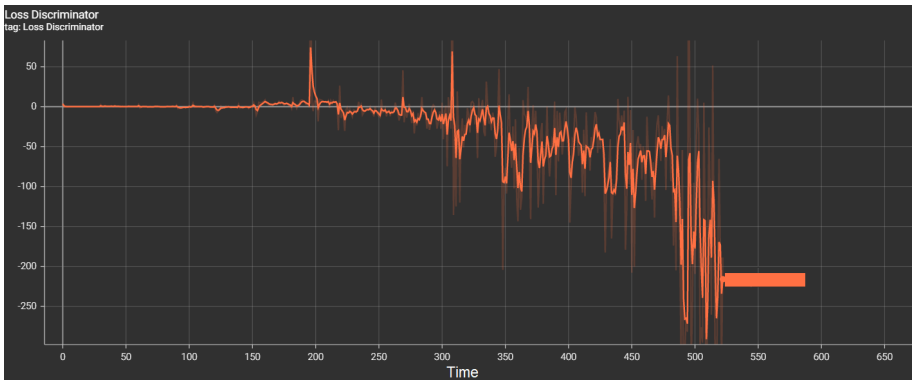


Figure 5.2: This image illustrates one of the reason of instability, during the training. As the ProGANs grow from lower resolution to higher, it is easier for discriminator(D to classify the generated image. This trait makes D to overfit the system which hinder the learning of generator G. After some epochs, the generator makes the ProGans stuck which leads to null values for D

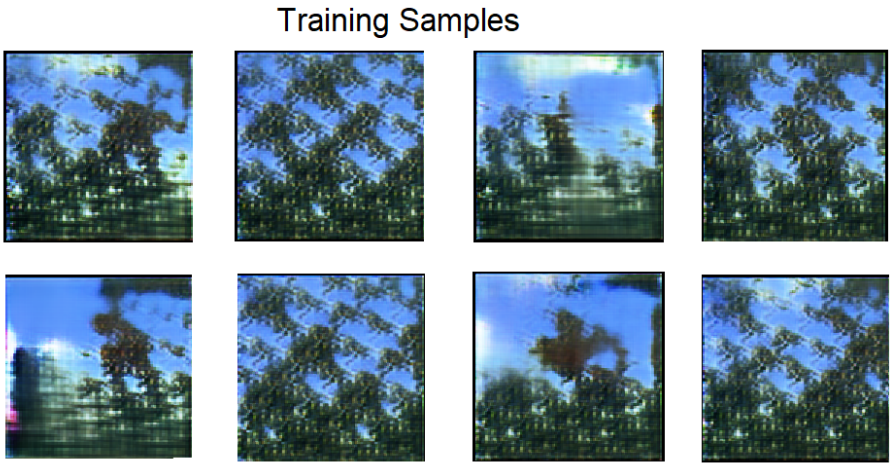


Figure 5.3: This image illustrates the generated results during the course of training. It can been seen upon inspection that quality of the generated images are being deteriorated.

to the custom dataset of apple orchard. Progressive GANs does not require paired dataset from domains as they are designed to produce synthetic data from a single source of image distribution.

During the development of this methodology, null values in the training, was constantly faced. Upon inspection, it was discovered that loss value from discriminator increased in range when the model trains at higher resolution. This phenomena breaks the training for ProGANs, however, before reaching this stage, the model has already stopped learning due to the higher difference between loss value. This can also be seen in the fig 5.3 The custom-made dataset is used for the training purpose with a volume of 11,000 images as compared to 800,000 images used in the paper by the researchers. Due to the small volume of the custom dataset, the training procedure is very unstable. Initially, the image resolution is 4×4 which consumes marginal GPU memory, however as the training goes, the network expands to higher resolution like 1024×1024 which results in the higher usage of GPU memory that leads to the problem of unstable training. During the development of the thesis, It was observed that higher resolution is not a requirement of the synthetic

data because many machine learning algorithm applications work with lower resolution images for training. A GAN must also learn to map the style from training data however pro-GANs lacks this functionality. After many trials and errors, it was decided not to pursue this methodology.

It can be observed in fig 5.1 that progressive GANs failed to reproduce the results with good quality. It can be seen the background of the input images was successfully created by this model however it has failed miserably in the generation of the leaves and apples. The computed FID score from these generated images is very higher which indicates the bad quality of the images.

5.2 Cyclic GANs For Synthetic Data

After the failure of Progressive GANs, it was a requirement to use other variants of GANs which must produce low FID score results. The cyclic GANs can translate the image from one domain to another domain while keeping spatial integrity. A semi-paired dataset of the apple orchard field is used for the training of cyclic GANs. This semi-paired data consists of two domains: one is designed in the unreal engine and another domain is from the real world. The cyclic model has produced high-quality results with very stable training. The semi-pared dataset contain two domains of apple images. One domain belongs to 3D apple orchard designed in the unreal engine whereas other domain belongs to real apple orchard. It was decided to make both domains semantically similar so mapping function will not find it complex to learn the mapping between domains. In the 3D apple orchard dataset, the simulated environment of apple orchard field is designed in unreal engine. This simulated environment has three day time settings. The global light source in the unreal engine are set behind the apple orchard field which gives the effect of early morning. The next location of light source is set right above the apple field which gives the effect of noon. When the light source is set at location of 180° approximately across the field then it gives the effect of evening The main reason for these settings is to variation in global light source. This way, it is feasible to set location of the light source in simulated environment, which will cast different shadow and environment effects in the simulated images. All of this shown in the fig 5.4

Every GAN model has tendency to introduce unwanted artifacts in the generated images. This happen normally due to defects in the underlying dataset or in the model's architecture. There are many procedures available which gives the remedy to this problem. Cyclic consistency is the solution provided by the researchers of the cyclic GANs. In this technique, the generated image must be approximately similar to the original input image. The input image comes from the dataset so it is important to create imaging dataset which without any artifacts.

In this experiment of variable locations related to global light source, in the simulated environment. it was discovered that it has drastic effects on the generation of the images by cyclic GANs. It was also observed that quality of real imaging data of apple orchard field, can impose significant effects in the output. The light incident of the sun in real images also casting shimmering effects in the real imaging data of apple orchard.The color of the apples in real images are also green which mix with the green leaves of trees which makes it difficult for the filters in model to distinguish between apples or leaves. When there is irregular or very complex changes in the spatial information of real apple

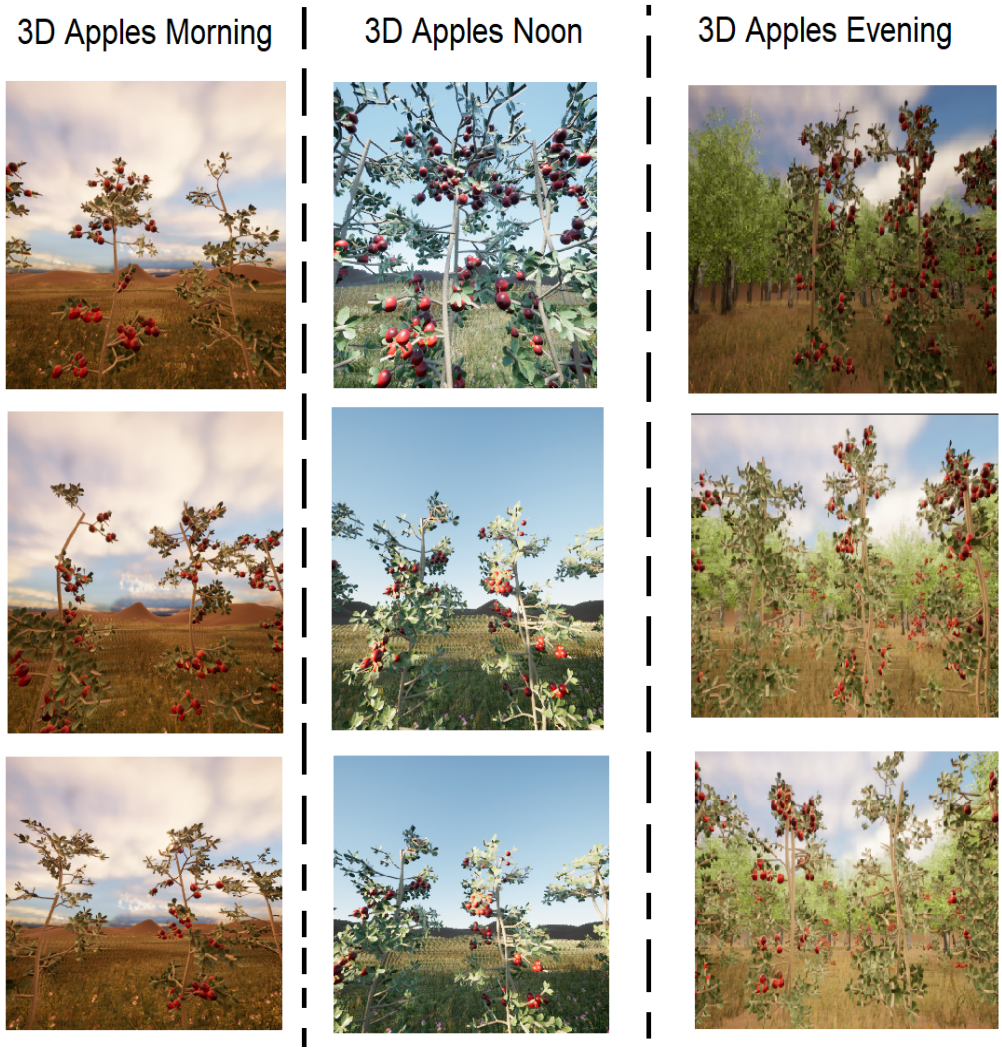


Figure 5.4: This image illustrates three daytime morning, noon, evening in the simulated environment provided by unreal engine for 3D apple orchard.

image then it is difficult for the mapping function to learn the mapping of such change between the domains. After the subjective visual inception, it was noted that the quality of the generated images from cyclic GANs is not very superior. In the fig 5.5, it can be observed the cyclic GANs struggles to learn the mapping from 3D images in morning settings to Real images. The generated results lack clarity or crispness of the apple tree. This synthetic apple images is not suitable as legitimate training data. The computed FID scores also confirms this hypothesis as the scores ranges between 200 to 250.

Similar experiment is performed for in the noon settings in which the 3D images have least presence of shimmering or shadow effects. However, since real images contain actual shadow and shimmering effects due to the location of the sun, the final results are not very high quality. In the noon setting, the light source is almost at 90° angle of the 3D apple orchard in unreal engine. The distance between the camera and the apple trees remains similar, throughout the experiments. In this thesis, it in important to observe the effects created in the 3D images, translated into to generated images because cyclic GANs also provide style transfer from first domain to second domain. It can be observed

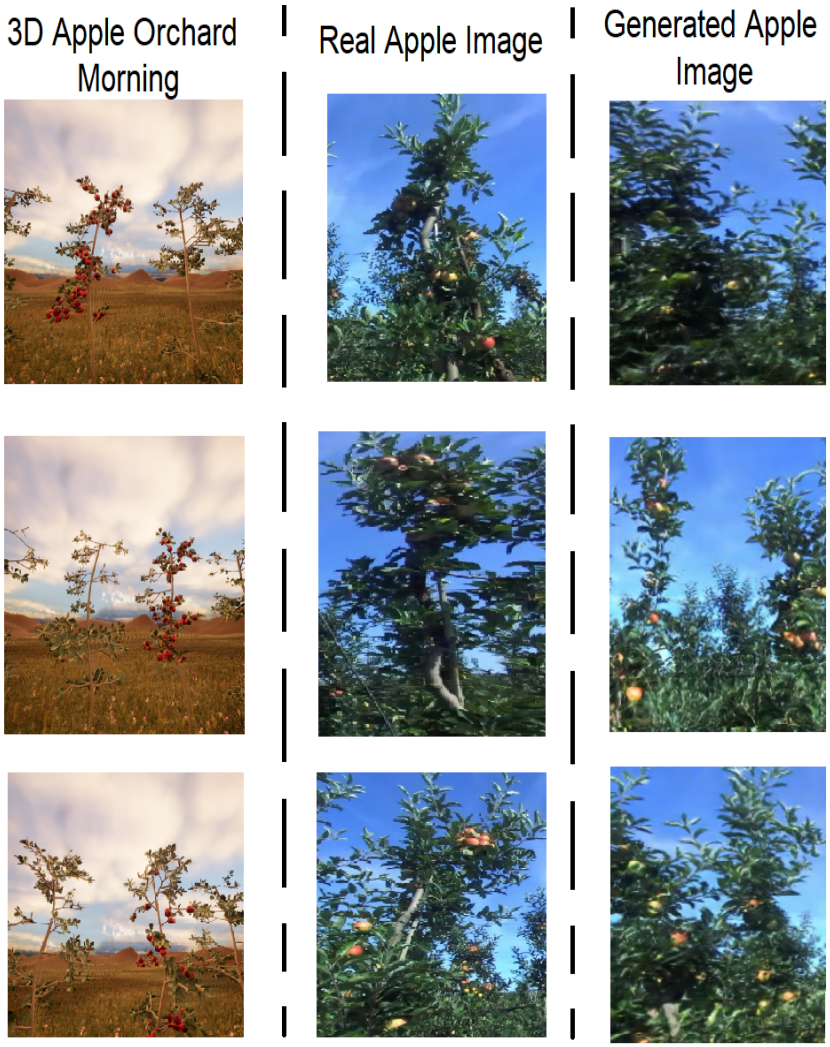


Figure 5.5: In this figure, for 3D apple orchard, Morning setting is chosen as fist domain in the training data. The cyclic GANs has successfully translated the global mapping between two domains however, in internal translation, the leaves and apples are not properly generated which gives these GANs higher FID score.

in the fig 5.7 that the mapping function of cyclic GANs did not produce high quality results. However, in the noon setting of 3D apple images, there are minimum shimmering or shadow effects. It can be observed that quality of the Real imaging data distribution plays a very important role for the GANs to generate high quality images. If the image quality of any domains hinders then mapping function will have difficulty to learn the RGB image channel data mappings.

As it can be observed in fig 5.6 that this experiments does not effect the quality of the generated images from cyclic GANs. It can be seen that cyclic GANs have learned the higher hierarchical features like the difference between blue sky and green trees, however, lower hierarchical features like intrinsic spatial information of leaves and apples, are difficult to learn. The generated outputs exhibit similar lower quality aesthetics. The geometric information in the real apple imaging dataset,is feasibly complex for the GAN to learn. The cyclic GANs are designed to mimic the underlying image data distribution during the

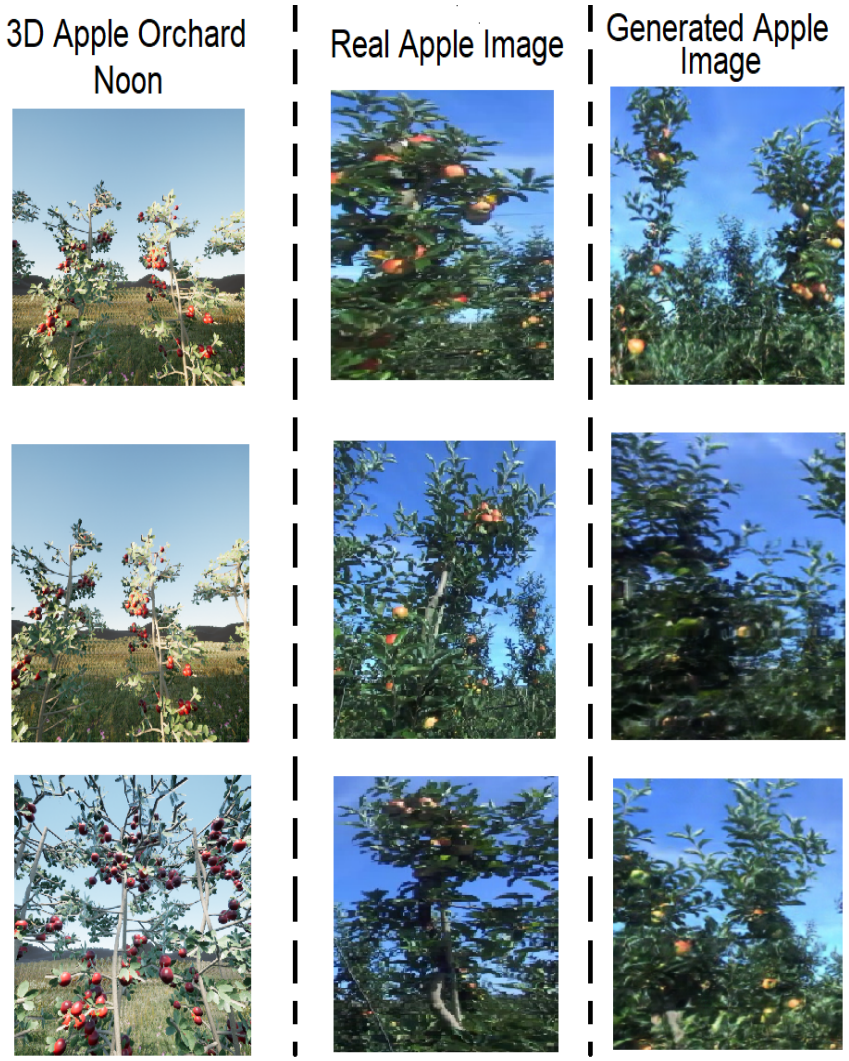


Figure 5.6: In this figure, for 3D apple orchard, noon setting is chosen as fist domain in the training data. Upon close inspection, the placement of leaves and apples are not correctly generated at logical place in the images which gives bad quality score to these outputs.

training process. They sometimes succeed to full map the probability distribution of the RGB channel data in the images. This is the subjective measure to tell how much mapping has been learned. Sometimes GANs learn full mappings, sometimes their learning is not adequate enough to produce training data related images with high variance. However, they are designed to produce anything they have learned from the data distribution. It is subjective to the end users of this generated synthetic data for use or not. The GANs are very complex systems which are designed to learn the probability distribution of the information available in the imaging data. This ability makes them very unpredictable for the generation of the data. It can be observed in the fig 5.7 that generated images does not improve in the quality after the change of light source, in the 3D imaging apple data. The computed FID scores of this experiments in table5.2, with all three daytime settings proves that generated images are not of high quality.

In the next experiment, different real dataset is chosen to observe the changes in the generated images. In this Real images, there is no direct sun light on trees, which helps in

Figure 5.7: In this figure, for 3D apple orchard, evening setting is chosen as fist domain in the training data. Again the quality of the generated outputs is not very good. The cyclic GANs has successfully generated backgrounds and foregrounds, however, they failed to add details in the foreground.

FID Scores of Cyclic GANs	
Datasets	FID scores
3D Apple Images Morning, Real Apple Images	200.65141269
3D Apple Images Noon, Real Apple Image	240.98094322
3D Apple Images Evening, Real Apple Image	230.7891123423

Table 5.2: This table shows the FID score of the Cyclic GANs trained in three daytime settings. The higher scores informs about lower quality of the generated images. It can also be related to the figures 5.5,5.6, 5.7

the reduction of shimmering and shadow effects. It can be observed in the fig 5.8 that final generated results are not very crisp or detailed. For the training of cyclic GANs, three datasets were selected from three different daytime. It has been observed that the quality

of the final generated images degraded due to less number of training data selected. For this test, number of apple images belong to real dataset, is 978 and 1000 images from 3D apple imaging distribution. In cyclic GANs, volumes of the training dataset have drastic effect on the final result. After examining the results in fig 5.8, it is concluded that volume

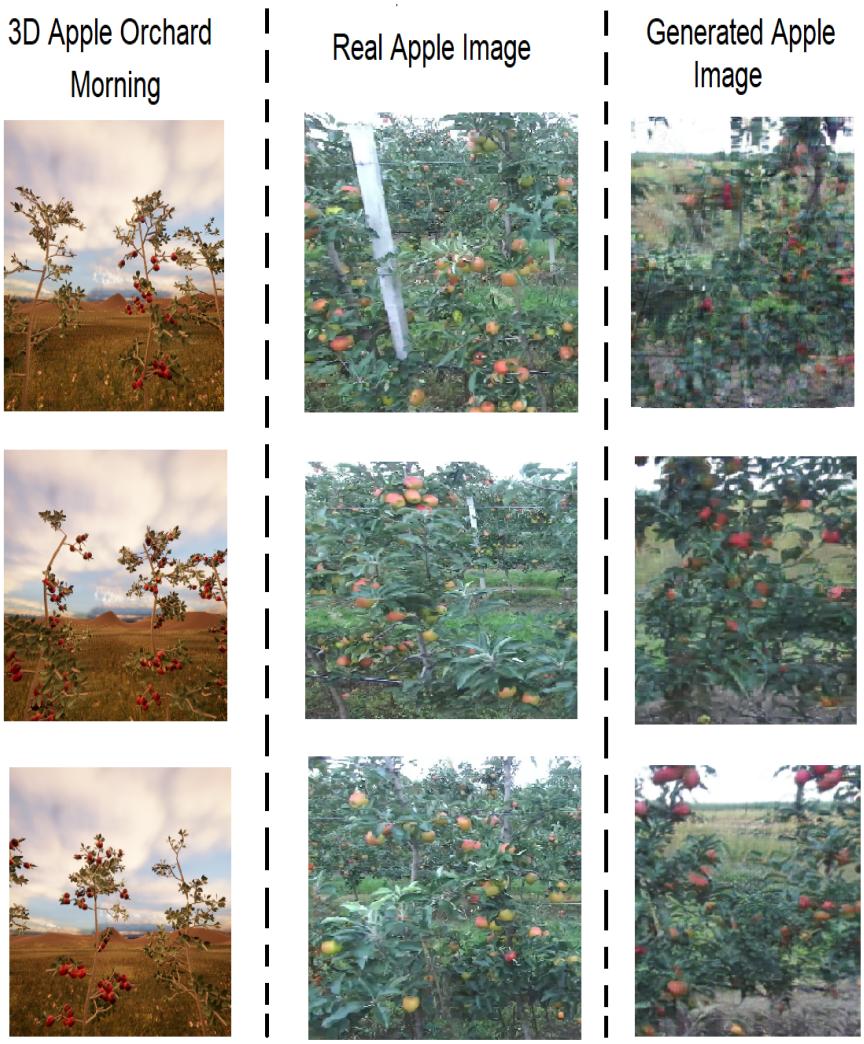


Figure 5.8: In this figure, for 3D apple orchard, morning setting is chosen as fist domain in the training data. Upon closer inspection, the locations of apples and leaves have generated at proper logical places, however, they are not very detailed.

of training dataset must be adequate enough which allows the mapping function of the model to learn the internal RGB channel data distribution. It is must be noted that from naked eye, the results seem good but upon close inspection, the degraded apples and leaves can be observed which hamper the overall quality of the final result. it can be seen in the fig 5.9 when the noon daytime setting was chosen, the cyclic GANs could not produce high quality results. Even the change in the direction of the global light in 3D images of Apple orchard, did not produce the drastic effects in the final images. Although, the 3D dataset in this light setting has maximum shimmering and shadow effect. The underlying reason remains the same as there not enough volume of the dataset, from which a mapping function can learn the the mapping.

Figure 5.9: In this figure, for 3D apple orchard, noon setting is chosen as fist domain in the training data. The apples in generated images are less sharps which gives bad FID scores.

In the next experiment, the evening daytime is selected for the observation of final generated images by cyclic GANs. It can be observed that evening daytime settings did not impose any drastic or noticeable results in the final generated results.It must be remembered that the both morning and evening daytime exhibits similar light effect in the 3D images. So the final results from this daytime is expected to be similar like morning daytime. Throughout the experiment time, it was noted that 3D apple orchard data does not impose noticeable changes to the generated images until the quality of the second domain improved. A GAN must learn to map probability distribution of data from both domains and any disturbance in any domain can hamper the mapping. As it can be observed in the 5.10 that even the change in the global light source did not have affect on the generated images. It can also be seen on the number produced by the FID scores,in table 5.3. It can be seen that small volume of the training images have produced higher FID scores which interprets the lower quality of the generated images. In this experiment, the higher FID scores corresponds to the marginal quality images which is the result of lower volume in the training data. In the coming experiments, volume will be increased in the training data for the betterment

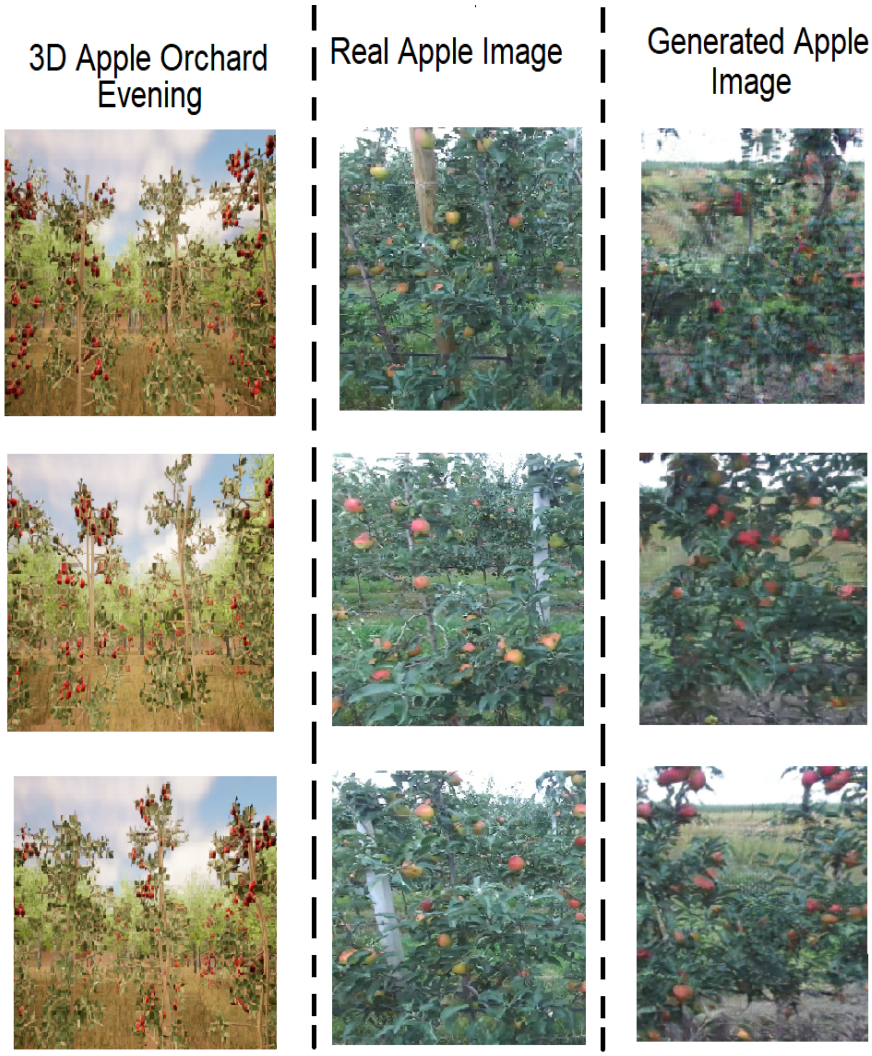


Figure 5.10: In this figure, for 3D apple orchard, evening setting is chosen as fist domain in the training data. The generated output gives global impression of apple tree but after close inspection the apples appeared to have less detail information as compared to real training images.

in the quality of images.

FID Score of Cyclic GANs	
Datasets	FID scores
3D Apple Images Morning, Real Apple Images	180.41269893
3D Apple Images Noon, Real Apple Images	190.6120909432
3D Apple Images Evening, Real Apple Images	200.234234591

Table 5.3: This table shows the FID score of the Cyclic GANs trained in three daytime settings. The higher scores informs about lower quality of the generated images. These scores are also reflected in 5.8,5.9,5.10

A very important aspect will be discussed in this experiment when the sun light is behind

the real apple trees, creating dark effect over the images. The number of images are comparatively very high but,however, due to less availability of the spatial information in the real data distribution, it is difficult for the mapping function to extract the needed information from the real images. In the fig 5.11, The cyclic GANs is facing the issue

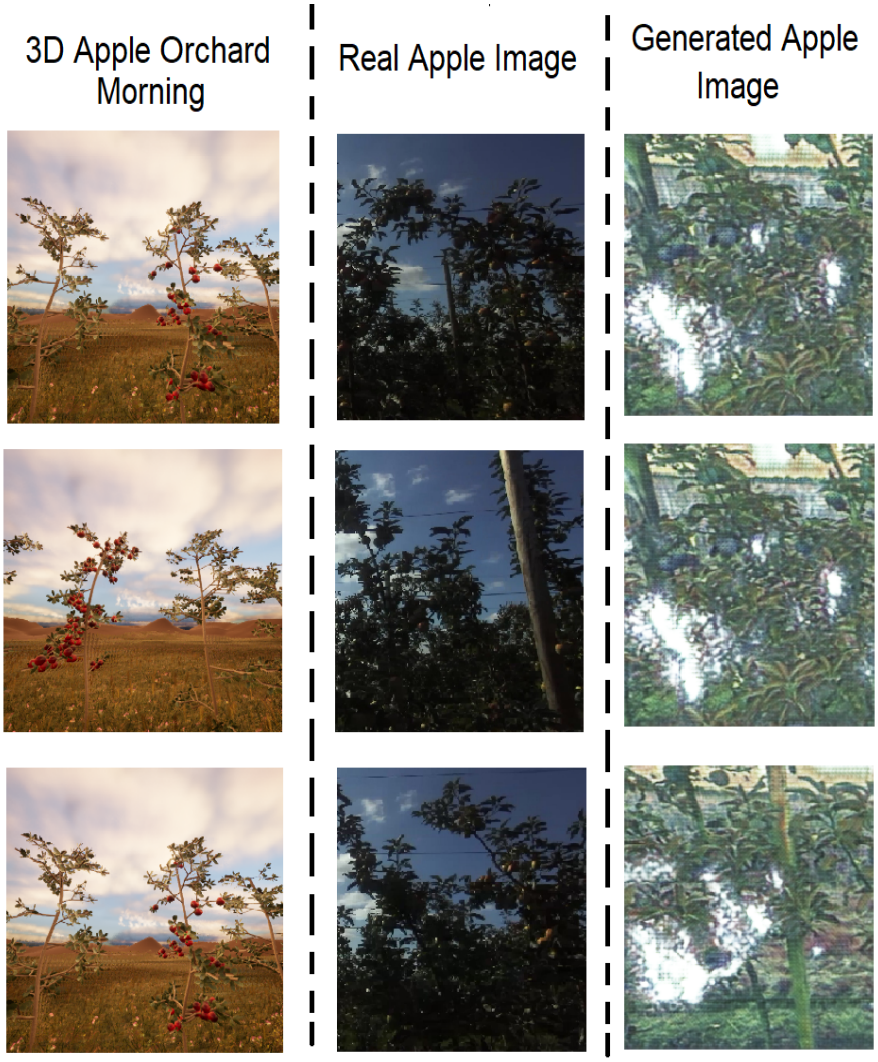


Figure 5.11: In this figure, for 3D apple orchard, morning setting is chosen as fist domain in the training data. The real imaging apple orchard data contains back-lighting issue which darken the foreground region of the image.

of Back-lighting. In the back-lighting, the main source of light is behind the object which creates darker regions in the image. These darker regions can hide the necessary information which GANs needs to learn for the regeneration of the images. A similar issues has been faced in real imaging apple orchard data where the sun light is behind the apple trees. The generated results can be seen in the fig fig 5.11 where mapping function of the cyclic GANs is facing extreme challenges to learn the RGB channel data distribution in real images. It must be noted the images from 3D apple orchard, in morning setting, are not creating significant effect on the final results. It can be observed in the fig 5.12 that even change in the source light does not have any impact on the final generated image. It must be noted here the volumes of training data is above 10,000 images. This

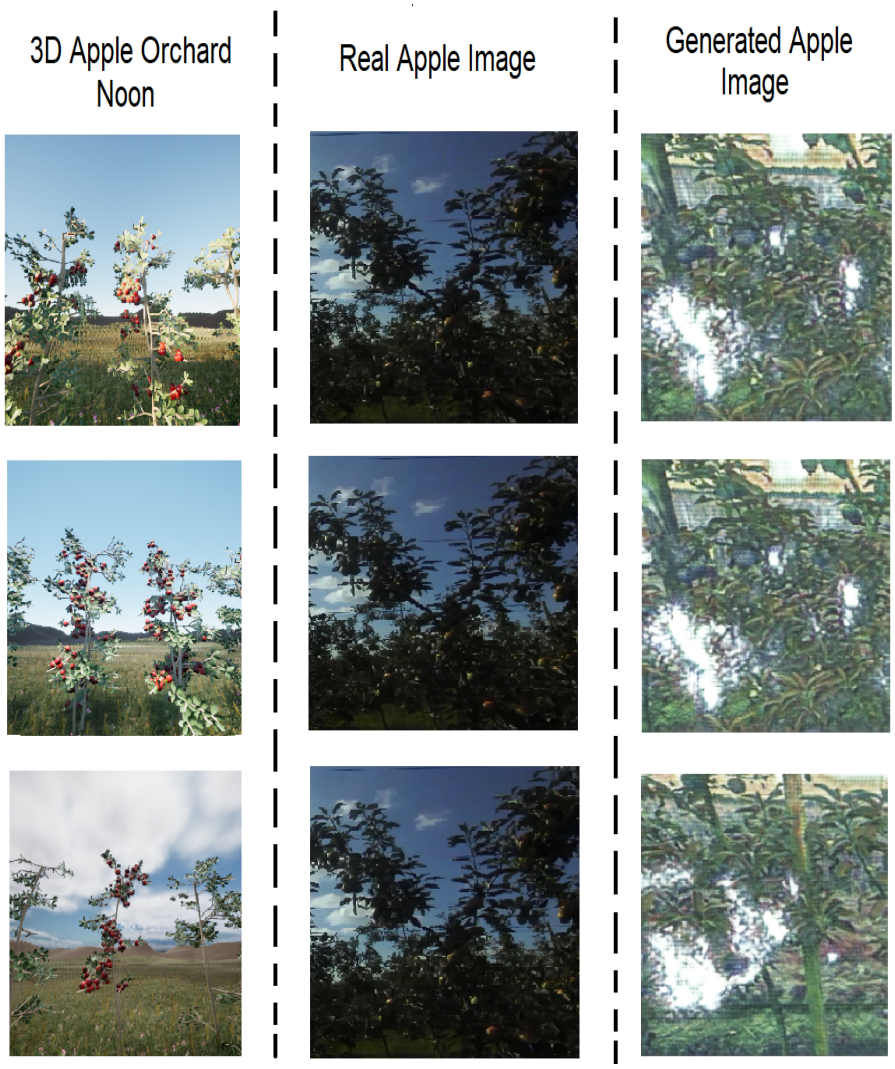


Figure 5.12: In this figure, for 3D apple orchard, noon setting is chosen as fist domain in the training data. The real imaging apple orchard data contains back-lighting issue which darken the foreground region of the image.

experiment was conducted to inform about the issue of back-lighting in the training data. For the cyclic GANs, the real images from apple orchard field seems like dark images which contains noticeably less information than predecessors. Any change in location of the global light source in 3D images, cannot ensures the higher quality for the generated images. A mapping function has fewer information to map from 3D apple orchard to real apple orchard domains. the immediate remedy of this problems is to whiten the darker regions of back-lighted images. This way mapping function of cyclic GANs can see more information for generation of the images. But underlying assumption says that only 3D apple orchard images goes through artificial remedies since they belong to the simulated environment in unreal engine. Real images must not go through the any artificial photo enhancements procedures. In this fig 5.13, the morning daytime settings has no substantial effect on the final generated images. The cyclic GANs fails to replicate the data distribution from real apple orchard data. It must be noted here that volume of the training data, which contain two sets of image where one set belongs to 3D apple orchard

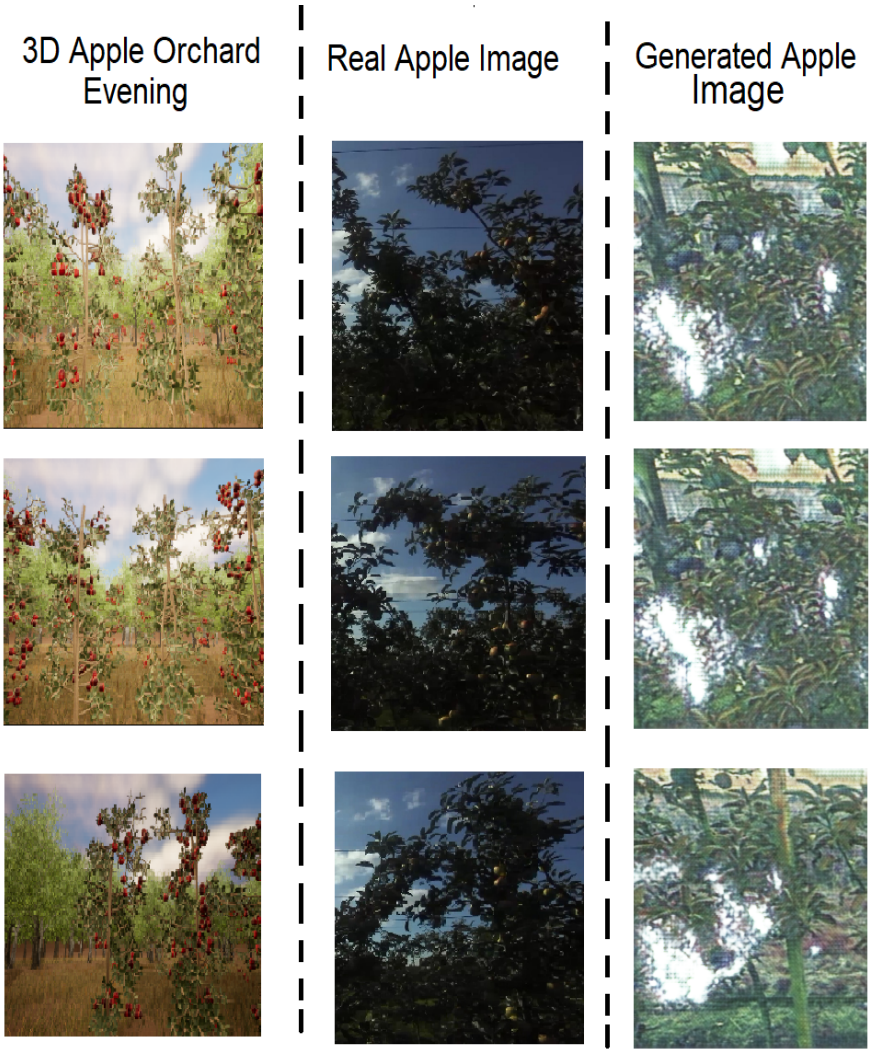


Figure 5.13: In this figure, for 3D apple orchard, evening setting is chosen as fist domain in the training data. The real imaging apple orchard data contains back-lighting issue which darken the foreground region of the image.

image distribution and second set belong to real apple orchard distribution,is around 10,000 images. The computed FID score also tells the similar story about the quality of the final images generated through cycle GANs. The higher scores in the table 5.4 tells the

FID Score of Cyclic GANs	
Datasets	FID scores
3D Apple Images Morning, Real Apple Images	302.41269893
3D Apple Images Noon, Real Apple Images	310.6120909432
3D Apple Images Evening, Real Apple Images	340.234234591

Table 5.4: This table shows the FID score of the Cyclic GANs trained in three daytime settings. The higher scores informs about lower quality of the generated images and these scores are also reflected in the generated images in 5.11,5.12,5.13.

similar story about the quality of the images. In this experiment, it is observed that light source in both domains plays very important role in the final generation of the synthetic images of apple orchard.

In the next and last experiment, there are multiple parameters that are taken care of in order t create good quality synthetic data. In the 3D apple orchard dataset , the global light source is set for three location in order to simulate the three daytime like morning, noon and evening. For the real apple orchard dataset, no back-light effect exists in the images of apple orchard field. There is no shimmering or shadow effect present in the images. It can bee seen in the 5.14 that generated results shows very high quality. Apart

Figure 5.14: In this figure, for 3D apple orchard, morning setting is chosen as fist domain in the training data. The real imaging apple orchard data also contains no effects which allow GANs to produce higher quality results.

from the resolution difference between real and generated data, both images from domains are indistinguishable. It must be noted here that volume of training data is around 11,000 images. This gives cyclic GANs enough space to learn the probability distribution of the mapping between 3D and real imaging dataset. Upon close inspection, it can be observed that red color of apple fruits has been translated from 3D apple orchard to generated

apple images. It can be observed in the fig 5.15 that noon daytime setting in the 3D apple

Figure 5.15: In this figure, for 3D apple orchard, noon setting is chosen as fist domain in the training data. A real data of apple orchard contains real images. A generated images are from cyclic GANs which learnt from 3D apple noon and real apple domains. The finer details of apples, leaves and wood can be observed with naked eye which reflects higher quality.

orchard data contains shimmering and shadow effects, however, it did not effect the quality of the final result. The mapping function learn the underlying features from the 3D apple orchard dataset and translated these features to the generated image. The image-to-image translation occurs when cyclic GANs learns the feature from both domains and map them into the generated image plane, along with very high variance. It must be noted here that both the domains are semantically equivalent which helps the mapping function learn internal mapping without any drastic geometric spatial change. This assumption has been very successful in the generation of the very high quality images. It is very important for the cyclic GANs to have adequate volume of training data for the training procedures. This considerable volume of the data contains adequate information for the GANs to learn with 200 epoch iterations. In each iteration the model corrects its self through cyclic consistency loss. In the fig 5.16, it can be observed that generated image

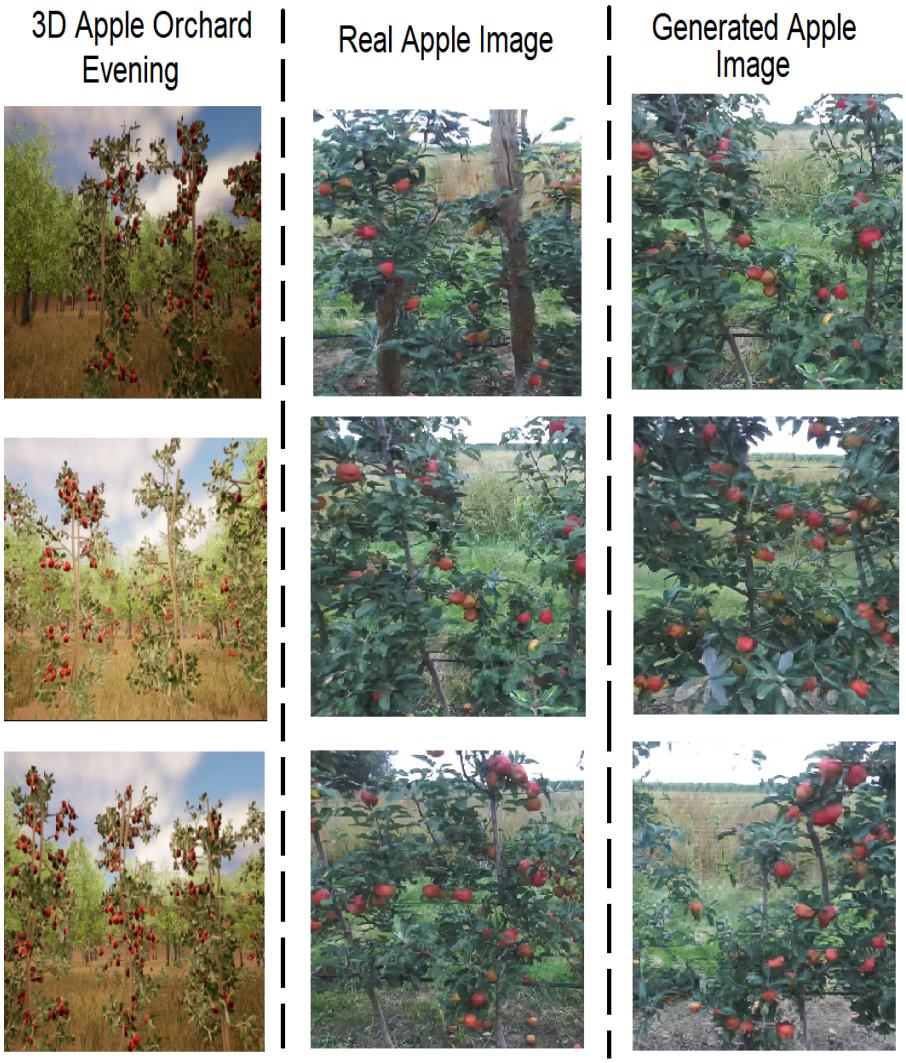


Figure 5.16: In this figure, for 3D apple orchard, evening setting is chosen as fist domain in the training data. A real data of apple orchard contains real images. The generated samples have detailed information of apples and leaves which gives lower FID score.

quality has significantly increased. The computer FID scores in table5.5 also corresponds to the higher quality results. As it mentioned earlier that lower the FID scores mean higher quality images are generated with cyclic GANs. Upon close visual inspection, it can be seen that generated images from cyclic GANs contains adequate amount variance which means every image is semantically different but spatially similar to the next image. The image-to-image translation through cyclic GANs have proved to be viable for the generation of synthetic data related to apple orchard. In this thesis, a dataset, with the combination of two domains have proved to very good for the training. An idea was to develop 3D images of apple orchard field, in simulated environment which will be used for the style transfer to the real apple imaging dataset. Every GANs have some limitations due to their complex two game architecture. This fundamental architecture design makes GANs very difficult to optimally train for the high quality generated images.

FID Score of Cyclic GANs	
Datasets	FID scores
3D Apple Images Morning	28.89341269
3D Apple Images Noon	24.61209432
3D Apple Images Evening	26.091123423

Table 5.5: This table shows the FID score of the Cyclic GANs where the lower score shows the high quality of the generated image under different light conditions. The FID scores are also reflected in 5.14,5.15,5.16

5.2.1 Limitations of Cyclic GANs

Every deep learning model has limitations that can hinder the performance of the network. In the case of cyclic GANs, it can be observed that when target and source imagine data distributions are not semantically equivalent then learned mapping can generate out-of-shape objects within the image. However, this method often gets successful when it has to learn the mapping of color or texture between two distributions. It has been observed through visual inspection of generated output that cyclic GANs are not able to segment between foreground and background which leads to unwanted anomalies in images.It has also been noted that cyclic GANs are suitable for only global image style transfer, however they fail in object transfiguration. The tasks related to geometric transfiguration are difficult for the cyclic GANs to accomplish as it can be observed in the fig 4.26 where the mapping between a human face and Ramon Food is not very success full. This is caused by the generator architecture which is designed to give quality performance when dealing with appearance changes rather than geometric changes. When the real images of apple orchard have back-lighting problems then the quality of the generated images drastically decreased. It is also noted that when the volume of the training data is reduces, training procedure does not converge well. The remedy of this problems is to have significant volume of dataset for the model to learn the mapping. It has been noticed through experiments that amount of image data must be approximately equal between both domains. There are cases when the volume of the 3D apple images, is less than 80% of the real imaging data which result in the lower quality of the generated images.

6. Conclusions

This chapter presents the summary of the approaches, experiments and results along with some suggestion for the improvement in the performance.

6.1 Summary and Discussions

Machine learning tasks need data for supervised or unsupervised. Good quality data results in good quality output. Agriculture is a field where imagine data is scarce. Many research companies are investing huge capital for the collection of good crop images. Researchers then turned to deep learning image-to-image translations methodologies to fulfill the demand for high quality imaging data related to agriculture. To achieve this task, a case study had been prepared to check the validity of the deep learning model, trained on synthetic data. Due to this study,the decision was taken to design a 3D model of the sugar beets in a blender and place them in an unreal real's stimulated farm-like environment where the inbuilt camera takes the images of planted sugar beets. The synthetic data is used for the training of a deep learning classification algorithm. The task of the algorithm is to classify the age of sugar beets while their growth period. For the classification purpose, only the real dataset of sugar beet is used. At the end of this case study, it was concluded that synthetic data can be deployed for machine learning tasks. To answer the question of unavailability of the varied dataset related to the field of agriculture and usage of synthetic data with style transfer, two deep learning GANs were deployed. For any machine learning-related tasks, the formation of a dataset plays a cerebral role, therefore custom data set was created where the source domain consists of 3D images of the apple orchard field in an Unreal environment, and the target domain consists of real images from apple orchard field. There are three daytime like morning, noon and evening in simulated environment which ad generalization in the 3D training data. First deep learning GANs which are utilized by this research are called Progressive GANs. This methodology is based on an iterative accumulation of layers from lower resolution to higher resolution and it is designed to produce higher resolution images with stable training. However, ProGANs are prone to instability due to the very complex nature of the training methodology. The results produced by this network are below any standard of quality. FID score is mentioned in the table 5.1 where it is very high. After the conclusion of the ProGANs, it was decided

to use another state-of-art image-to-image deep learning-based translation method. This method is designed for the unpaired dataset where the mapping is learned through cyclic consistency. This method contains two generators and two discriminators which allows to regenerate the imaging data and compare it with original imaging data. The cyclic consistency loss is then drive the model towards convergence. It can be seen that Cyclic GANs have produced high-quality results. It can be observed in the table 5.5 that the FID score of Cyclic GANs has been significantly improved 80% over ProGANs results.

6.2 Future Work

This section discussed the future work related to the synthetic data generation for agriculture, through GANs. As mentioned earlier, GANs are going through extensive research in the A.I community. Their unsupervised learning nature sometimes generates fault results that need further investigation. In this thesis, it was attempted to utilize different variants of GANs to synthesize the image data of apple orchard. It has been observed that sometimes GANs produce faulty results or the semantics of the image do not match the source image. Object transfiguration is also the regular issue that has been faced by this thesis. To overcome these issues, the architectural design of the generator must be changed to generate only agricultural-related images. In this architectural design, more convolution layers must be added along with residual layers which allow the model to capture intrinsic details of the spatial information available in the dataset. It must be noted that with the help of experiments, the volume of dataset must be adequate enough for the model to learn the mapping of imaging apple data domains. In the 3D images of apple orchard, if the modeling of simulated environment improved then the quality of the final generated images can be significantly increased. With the help of professional 3D environmental designers, it is possible to create high resolution high detailed 3D images of apple orchard. The high quality 3D renders of apple orchard environment can be achieved through the use of latest 3D tools like Unreal Engine 5. The higher resolution images from 3d apple orchard and real apple orchard can have very good impact on the final quality of the results. For real imaging data, high end DSLR cameras can be used better quality images, however, this higher resolution dataset requires significant volume of in-memory space and processing power for successful training of the model. After the architectural changes in the cyclic GANs, it must be noted that training procedures require high performance systems like VG100 GPUs. These high systems with adequate availability of in-memory storage, can speed up the training time. Cyclic GANs are designed for general image-to-image translation between semantically different domains. The network can be specifically tailored for the generation of complex spatial data in images like plants, trees. The agricultural based image generation through GANs has seen little success among the researchers, however, this thesis has created new opportunities for the application side of cyclic GANs.

Bibliography

[Balan 18] P. S. Balan, L. E. Sunny, "Survey on feature extraction techniques in image processing", *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 6, no. III, 2018.

[Barua 19] S. Barua, X. Ma, S. M. Erfani, M. E. Houle, J. Bailey, "Quality evaluation of gans using cross local intrinsic dimensionality", *arXiv preprint arXiv:1905.00643*, 2019.

[Bradski 00] G. Bradski, "The openCV library.", *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, vol. 25, no. 11, pp. 120–123, 2000.

[Cai 21] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, Y. Pan, "Generative adversarial networks: A survey toward private and secure applications", *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–38, 2021.

[Chiu 20] M. T. Chiu, X. Xu, Y. Wei, Z. Huang, A. Schwing, R. Brunner, H. Khachatrian, H. Karapetyan, I. Dozier, G. Rose et al, "Agriculture-vision: a large aerial image database for agricultural pattern analysis. 2020 IEEE", in *CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE*. 2020, pp. 2825–2835.

[Deng 09] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, "Imagenet: A large-scale hierarchical image database", in *2009 IEEE conference on computer vision and pattern recognition*, Ieee. 2009, pp. 248–255.

[Denton 15] E. L. Denton, S. Chintala, R. Fergus et al, "Deep generative image models using a laplacian pyramid of adversarial networks", *Advances in neural information processing systems*, vol. 28, 2015.

[Doersch 16] C. Doersch, "Tutorial on variational autoencoders", *arXiv preprint arXiv:1606.05908*, 2016.

[Efros 99] A. A. Efros, T. K. Leung, "Texture synthesis by non-parametric sampling", in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2, IEEE. 1999, pp. 1033–1038.

[Goodfellow 14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, "Generative adversarial nets", *Advances in neural information processing systems*, vol. 27, 2014.

[Gulrajani 17] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. C. Courville, "Improved training of wasserstein gans", *Advances in neural information processing systems*, vol. 30, 2017.

[Han 16] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz, W. J. Dally, "EIE: Efficient inference engine on compressed deep neural network", *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 243–254, 2016.

[Hertzmann 01] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, D. H. Salesin, "Image analogies", in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 2001, pp. 327–340.

[Heusel 17] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium", *Advances in neural information processing systems*, vol. 30, 2017.

[Isola 17] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, "Image-to-image translation with conditional adversarial networks", in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1125–1134.

[Johnson 16] J. Johnson, A. Alahi, L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution", in *European conference on computer vision*, Springer. 2016, pp. 694–711.

[Karras 17] T. Karras, T. Aila, S. Laine, J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation", *arXiv preprint arXiv:1710.10196*, 2017.

[Karras 19] T. Karras, S. Laine, T. Aila, "A style-based generator architecture for generative adversarial networks", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 4401–4410.

[LeCun 15] Y. LeCun, Y. Bengio, G. Hinton, "Deep learning", *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[Liakos 18] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, D. Bochtis, "Machine learning in agriculture: A review", *Sensors*, vol. 18, no. 8, p. 2674, 2018.

[McCarthy 90] J. McCarthy, E. A. Feigenbaum, "In memoriam: Arthur samuel: Pioneer in machine learning", *AI Magazine*, vol. 11, no. 3, pp. 10–10, 1990.

[Meshram 21] V. Meshram, K. Patil, V. Meshram, D. Hanchate, S. Ramkteke, "Machine learning in agriculture domain: a state-of-art survey", *Artificial Intelligence in the Life Sciences*, vol. 1, p. 100010, 2021.

[Odena 17] A. Odena, C. Olah, J. Shlens, "Conditional image synthesis with auxiliary classifier gans", in *International conference on machine learning*, PMLR. 2017, pp. 2642–2651.

[Rosales 03] R. Rosales, K. Achan, B. J. Frey, "Unsupervised image translation.", in *iccv*. 2003, pp. 472–478.

[Rumelhart 85] D. E. Rumelhart, G. E. Hinton, R. J. Williams, "Learning internal representations by error propagation", California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[Salimans 16] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, "Improved techniques for training gans", *Advances in neural information processing systems*, vol. 29, 2016.

[Sanders 16] A. Sanders, *An introduction to Unreal engine 4*, AK Peters/CRC Press, 2016.

[Srivastava 17] A. Srivastava, L. Valkov, C. Russell, M. U. Gutmann, C. Sutton, "Veegan: Reducing mode collapse in gans using implicit variational learning", *Advances in neural information processing systems*, vol. 30, 2017.

[Szegedy 16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, "Rethinking the inception architecture for computer vision", in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2818–2826.

[Thanh-Tung 20] H. Thanh-Tung, T. Tran, "Catastrophic forgetting and mode collapse in GANs", in *2020 International Joint Conference on Neural Networks (IJCNN)*, IEEE. 2020, pp. 1–10.

[Turmukhambetov 15] D. Turmukhambetov, N. D. Campbell, S. J. Prince, J. Kautz, "Modeling object appearance using context-conditioned component analysis", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 4156–4164.

[Tyleček 13] R. Tyleček, R. Šára, "Spatial pattern templates for recognition of objects with regular structure", in *German conference on pattern recognition*, Springer. 2013, pp. 364–374.

[Umbaugh 10] S. E. Umbaugh, *Digital image processing and analysis: human and computer vision applications with CVIPtools*, CRC press, 2010.

[Yu 14] A. Yu, K. Grauman, "Fine-grained visual comparisons with local learning", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 192–199.

[Zhao 17] J. Zhao, M. Mathieu, Y. LeCun, "Energy-based generative adversarial networks. arXiv. 2019 https://arxiv. org/abs/1609.03126.(accessed 17 July 2020). Paper presented at the 5th Int", in *Conf. on Learning Representations, ICLR 2017, Toulon, France, 24–26 April*. 2017.

[Zhou 18] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, J. Liang, "Unet++: A nested u-net architecture for medical image segmentation", in *Deep learning in medical image analysis and multimodal learning for clinical decision support*, Springer, 2018, pp. 3–11.

[Zhu 17] J.-Y. Zhu, T. Park, P. Isola, A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks", in *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2223–2232.