2023

Comparative Study of Generative Models for Text-to-Image Generation

Nazia Siddiqui
University of Windsor

Comparative Study of Generative Models for Text-to-Image Generation

By

Nazia Siddiqui

A Thesis
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science
at the University of Windsor

Windsor, Ontario, Canada

2023

Comparative Study of Generative Models for Text-to-Image Generation

by

Nazia Siddiqui

APPROVED BY:

M. Khalid
Department of Electrical and Computer Engineering

B. Boufama
School of Computer Science

I. Ahmad
School of Computer Science

January 19, 2023

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

The development of deep learning algorithms has tremendously helped computer vision applications, image processing methods, Artificial Intelligence, and Natural Language Processing. One such application is image synthesis, which is the creation of new images from text. Recent techniques for text-to-image synthesis offer an intriguing yet straight forward conversion capability from text to image and have become a popular research topic. Synthesis of images from text descriptors has practical and creative applications in computer-aided design, multimodal learning, digital art creation, etc. Non-Fungible Tokens (NFTs) are a form of digital art that is being used as tokens for trading across the globe. Text-to-image generators let anyone with enough creativity can develop digital art, which can be used as NFTs. They can also be beneficial for the development of synthetic datasets. Generative Adversarial Networks (GANs) is a generative model that can generate new data using a training set. Diffusion Models are another type of generative model which can create desired data samples from the noise by adding random noise to the data and then learning to reverse the diffusion process. This thesis compares both models to determine which is better at producing images that match the given description. We have implemented the Vector-Quantized GAN (VQGAN) - Connecting Text and Images (CLIP) model. It combines the VQGAN and CLIP machine learning techniques to create images from text input. The diffusion model that we have implemented is Guided Language to Image Diffusion for Generation and Editing (GLIDE). For both models, we use text input from the MS-COCO data set. This thesis is an attempt to assess and compare the images generated using text for both models using metrics like Inception Score (IS) and Fréchet Inception Distance (FID). The semantic object accuracy score (SOA) is another metric that considers the caption used during the image generation process. We compute and compare the results for each label in the MS COCO data set. We highlight the potential causes of why the models may not be able to generate images through analysis of the results obtained. Our experimental results indicate that the GLIDE model outperforms the VQGAN - CLIP for our task of generating

images from text.

AKNOWLEDGEMENTS

I would like to extend my sincere gratitude towards my supervisor Dr. Imran Ahmad for his patience, support, and professional guidance during my graduate studies. With his input, I was able to look at my research with a different perspective and a more critical eye. I want to take this opportunity to express my gratitude to my thesis committee members for their beneficial advice and suggestions for my thesis. I want to thank my internal reader, Dr.Boufama and my external reader, Dr. Mohammed Khalid, for their time, effort, and valuable feedback. Dr.Robin Grass for being on my committee at short notice and always being easy to access and willing to help.

I am greatly indebted to my parents for their support and words of inspiration throughout my education. I would also like to thank my sister, Farheen Siddiqui for always encouraging and supporting me. I would also like to mention my friends Asim, Karan, Puneet, Suraj and Liya for making university life an unforgettable experience for me. I humbly extend my thanks to the School of Computer Science and all concerned people who helped me in this regard.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

GAN	Generative Adversarial Network
DRAW	Deep Recurrent Attentive Writer
FID	Frechet Inception Distance
IS	Inception Score
NFT	Non-Fungible Tokens
VAE	Variational Autoencoder
MS COCO	Microsoft Common Objects in Context
SOA	Semantic Object Accuracy
MNIST	Modified National Institute of Standards and Technology
DC-GAN	Deep Convolutional Generative Adversarial Network
CLIP	Contrastive Language- Image Pretraining
CLIP-GLaSS	CLIP-guided Generative Latent Space Search
DDPM	Denoising Diffusion Probabilistic Models
DDIM	Denoising Diffusion Implicit Models
GLIDE	Guided Language to Image Diffusion for Generation and Editing
CUB-200	Caltech-UCSD Birds-200-2011
VQGAN	Vector Quantized Generative Adversarial Network
SOA-I	Semantic Object Accuracy - Image
SOA-C	Semantic Object Accuracy - Class

CHAPTER 1

Introduction

1.1 Overview

Generative models have recently gained attention for their use in producing fake images. The emergence of artificial intelligence (AI)-generated fake images, referred to as "deep fakes" presents several challenges such as developing synthetic images that look realistic, images with multiple objects, and reliable evaluation metrices that align with human judgment [13]. However, new technologies appear promising for image generation. Developing a system that can create images representing a given textual description inspired by how humans perceive is a significant step towards computer intelligence [13]. Prior to the development of generative models, the process of creating an image from text relied on image querying. Which involved selecting the best collection of images from an image database to illustrate text description. Recent advances in artificial intelligence and computer vision have facilitated the creation of images based on text descriptions. The goal of text-to-image synthesis is to generate images from textual descriptions. Research on text-to-image creation has sparked much interest due to its applications in art, marketing, business, and education, among others. Several frameworks and improvements have been proposed to produce more visually realistic images. The representation of an image as the numeric values of its pixels is termed as image data distribution. If there is an image with the dimensions m x n pixels. The image may be interpreted as a vector with m x n dimensions, which is high dimensional data distribution. Image data distribution of high dimension makes image creation a complex problem. The generated images should be visually

realistic and semantically accurate for adequate text-to-image synthesis. Semantic accuracy refers to the agreement between the content of the image and the text description. Generative modelling is when we provide the model with a description of what we want to generate, and the model returns an image. The model automatically learns from the input data and replicates it with variety and accuracy. Suppose we have a description using which we try to generate an image. The generated image will resemble but not be identical to the input sample image because the model uses input images to learn the image's representation. That is why the representation is unique each time and varies for different models.

There are different types of generative models like Autoencoders [16, 21], Generative Adversarial Network (GAN) [16], and Diffusion models [35] for text-to-image generation that several authors have introduced over the years. These models have been compared based on visual realism, diversity, and semantic alignment to understand which model works better at generating an image related to the text used for its generation [9].

1.2 Text-to-image Generation and its Applications

After being trained on image data, experiments have shown that producing fake but photorealistic images is feasible. Ramesh et al. [32] showed text-to-image generation based on an autoregressive transformer in terms of zero-shot learning. Zero-shot learning implies creating samples for the text input without being trained on the same input. Scaling can result in more accurate generalization compared to earlier domain-specific techniques and the approach achieves the best frechet inception distance [44] and the highest inception score [1] when qualitatively comparing samples from proposed model to those from prior work [32]. This approach is used by the popular text-to-image generation tool DALL.E [9], which became available for public use by OpenAI shortly after the diffusion model was introduced. The diffusion model has

been proven to generate high-quality images and give desirable characteristics like distribution coverage, a stationary training aim of creating images from text, and simple scalability [9]. With these improvements, they achieve a new state-of-the-art, outperforming GANs on a variety of metrics and data sets [9]. DALL.E was released in January 2021, followed by DALL-E 2, which was released later in November 2022. Meanwhile, diffusion models gained popularity in the vision community. OpenAI chose this method as the basis for DALL-E 2 because it uses simple image-denoising nets to reduce a convex regression loss instead of a minmax.

DALL.E and other generative AI picture tools are the latest innovation that venture capitalists have been eager to try out. The use of non-fungible tokens (NFTs), a kind of digital asset that can be in the form of digital art, has skyrocketed. NFT artworks are already fetching millions of dollars [4]. With enough imagination, people can create digital art using text-to-image generators, which can be utilized to create NFTs. Creating realistic graphics from natural language can enable people to produce rich and varied visual content. There has been impressive progress in the first few years since Mansimov et al. [24] started with modern machine learning approaches for text-to-image synthesis [32]. The approach has numerous commercial, educational, and artistic applications. One of the commercial applications is the expensive production of video games and animated films, in which many production artists are employed to perform very mundane tasks. Text-to-image models can create and colourize characters automatically by giving their descriptions. The tool can be used to generated images for books and for teaching which makes visual learning easier and more accessible. The authors of short story books can use the tool for creating images related to the stories. The artists can look up for inspiration for their art and use images that are original. Developers can generate images and use them for their websites or applications without having to spend money for images or any copyright issues. The image generation can also be used for creating medical image data set which can be used for training for tasks like detection of diseases. Some other uses can also be to add variation to already existing data sets, generate relevant images from chatbot interactions, and fulfill the scarcity of image data sets.

The application possibilities for text-to-image generators in the future are limitless.

1.3 Generative Models

In terms of a probabilistic model, a generative model specifies how a data set can be produced. We can generate fresh data by sampling from this model. For example, consider a data set that includes pictures of cars. We want to create a model that, after learning the fundamental principles governing a car's appearance, can create a brand-new image of a car that has never existed but still appears realistic. This requires a data set containing many sample images of the object that is to be created [12]. Each of the samples in the training data is referred to as an observation. Each observation is made up of a variety of features. For an image generating issue, the features are typically the various pixel values. The model trained on the observations is able to produce new collections of features that appear to have been produced by using the same set of rules as the original data. The resulting photos will consist of a new collection of pixels that have been rearranged so that the item is identifiable as the same object, but is not identical to the original observation. Given the vast number of possible pixel assignments and the small number of feasible image layouts, this is a challenging task for image synthesis. Additionally, a generative model should produce probabilistic results rather than predetermined outcomes [12]. If a model just does a fixed calculation, such as averaging the value of each pixel from the data set, it is not generative because it constantly produces the same output [12]. The model should be able to learn an approximation of the input distribution and then sample from it to produce new, separate observations that appear similar to the initial training set [12].

1.4 Thesis Objective and Contribution

Synthesizing images from text descriptions has lately gained attention as a research topic due to the development of generative models. There are versatile methods for

conditional image generation that have made major advancements in recent years in terms of visual realism, diversity, and semantic alignment [13]. However, there are still several issues that the area must address through additional research, such as making it possible to generate high-resolution photos with several objects and creating effective evaluation metrics that are correlated with human judgment. One of these challenges is being able to tell how closely the image we make matches the text we use to generate it. The performance of a model depends on the specific task we are using it for. The assessment of generative models is crucial to understand where further research is required. In this thesis, we are comparing to draw conclusion upon how the two types of models, GANs and diffusion models, perform when we generate images over the same textual data from the MS COCO [22] data set and compare them using different available metrices. This thesis complements other previous research by using a new metric, i.e., semantic object accuracy (SOA) [18], which to the best of our knowledge has not been employed to compare GANs and the diffusion model. The evaluation will also reveal how the models behave for different types of objects. We discuss and analyze the reasons why a generative model is able and not able to generate images for a certain object. The aim of this research is to see how the models perform and which one is preferable when it comes to generating images from text.

1.5 Thesis Organization

Rest of the thesis is organized as follows:

Chapter 2 describes the literature on evolving text-to-image generation algorithms. Chapter 3 describes our methodology in detail. Chapter 4 discusses the experiments we have conducted and their outcomes. Finally, Chapter 5 discusses some future directions and concludes our work.

CHAPTER 2

Related Work

In this section, we briefly describe several previous research that this thesis is built upon:

2.1 Overview of different Generative Models

There are four major categories to classify generative models. Which include GANs, variational autoencoders (VAE's), flow-based models, and diffusion models. We will be discussing all of them in this section.

2.1.1 Generative Adversarial Networks (GAN)

Generative Adversarial Networks (GANs) have a generator neural network that creates what should be a realistic image from noise or from some useful conditioning variable, such as a class label or text encoding. The success is determined by the discriminator, which classifies the images as either a real image or a false image produced by the generator. GANs have shown that it is possible to create fake but highly realistic images. To create images from text using GANs, the simplest way to train is to treat text, image pairs as joint observations and train the discriminator to distinguish between real and fake pairs [26]. If the model is trying to generate the sentence "A yellow car", here the conditioning information is the yellow colour of the car. Early in training, the discriminator disregards the conditioning information and dismisses samples from generator because they don't seem realistic. The generator must get better at aligning the images it creates, like the "car" in this example. As soon as

it becomes adept at creating convincing images, it must also learn to match those images to the conditioning data [34]. There are different ways each model handles the learning of this conditioning information. A common machine learning task is to learn a density model, i.e., generating an estimate of a distribution based on observed data. For the density model of any object 'X', it would accept some input and say yes that's 'X' or no that's not 'X'. GANs have been used to learn density models. Zhang et al. [45] employed a collection of images with accompanying text labels as their training data, and the objective is to create a conditional density model by instead looking if there is a object 'X' given the condition 'Y'. Which allows us to define the features that is the conditional information that we desire in the resulting image.

For GAN to understand the text sentence, the model basically encodes the text description into an embedding which represents the text sentence and is used by the discriminator to identify 'real image with right text' , 'real image and wrong text' and 'fake image and right text'. The weights are adjusted accordingly until the generator is able to trick the discriminator into believing that the generated fake image is real. Again, the way the text is encoded is varied for each model and is significant for performance of the model.

Given the advancement and research on GANs, much efficient and improved models have been introduced that are far more effective at creating realistic-looking images from text. Reed et al. [34] introduced Stack GAN, demonstrating that their model could generate a photorealistic image from any text sentence. The model consists of two stages. In the first stage, it takes a sentence as input and outputs an image with primitive shapes and basic colours creating a low-resolution image. The stage 2 of GAN takes that low-resolution image and the original sentence as input and generates a much higher resolution version of that image by filling in all the details.

2.1.2 Variational Autoencoders (VAE)

A variational autoencoder (VAE) offers a probabilistic way to describe an observation in latent space. They take the input and encode it, often compressing it to a latent space of lower dimensionality [29]. The main goal of autoencoders is to efficiently

represent the data. Their task is to identify a low-dimensional representation of a high-dimensional input that enables reconstruction of the original input with little content loss. If the image is 784 pixels in size (28 x 28 pixels) in grayscale, the autoencoder would discover a way to map the 784-dimensional space onto a 2D space so that the compressed picture would only need to describe the X and Y coordinates of a location on the map. The autoencoder would next attempt to recreate the original 784 pixels using only the X-Y coordinates as input. Making a low-dimensional representation that enables the autoencoder to recreate the original input is its task. This makes sure that the latent space only contains the input features that are most important for reconstructing the input and is free of noise and unimportant features. It must have two components: the encoder, which takes the input image and reduces it to a low-dim representation, and the decoder, which performs the opposite operation by creating the original-sized image from the latent representation [29].

2.1.3 Flow-based Model

The class of models known as "flow-based models", explicitly learn the data distribution. Flow-based models learn specific encoders and decoders: Similar to the encoding stage in autoencoders, they apply a transformation "f", parametrized by a neural network, to the data. However, the decoder is not a brand-new neural network that must independently learn the decoding procedure and it is the exact opposite of the function. With neural networks, it takes quite a few methods to obtain this invertibility of "f" [42].

2.1.4 Diffusion Model

The Diffusion model learns to establish a Markov chain of diffusion steps to gradually introduce random noise to data, and then they learn to reverse the diffusion process to create desired data samples from the noise [42]. In contrast to VAE or flow models, diffusion models are trained using a predefined process, and the latent variable has a high degree of dimension [42]. A Markov chain is a set of variables where each

variable's state solely depends on the previous event. Using a Markov chain, the data is contaminated with random noise. Taking the image, we sequentially add a particular amount of noise to it during the forward diffusion phase. At each step the model generates the new image in the series by gradually increasing the noise and we repeat this a specific number of times until the image becomes noise. The diffusion process is reversed using a neural network. To create the image from diffusion step t to step $t-1$, the backward diffusion method uses the same network and weights at each stage. Instead of letting the network anticipate the image, one might decide to forecast the noise that has to be removed from the image at each step to further simplify the issue. In any event, the neural network's design must be chosen in a way that maintains the dimensionality of the data. There isn't much room for error when returning from noise to the original image by repeatedly denoising. Similar to GANs, the generation process passes through each of these checkpoints, adding more and more information to the image that was once just noise but compared to GANs, diffusion models are a more gradual, iterative, and controlled approach. The following iteration, which was introduced by Nichol et al. [26], successfully incorporated textual information into the generation procedure, enabling us to generate images using diffusion models from text. The data set consisting of images and their captions is used. The images became noisier and noisier after the forward diffusion process. Similar to the prior research from OpenAI, the diffusion model is trained to reverse this process using a UNet-based architecture. For the backward diffusion, the model takes the text caption into account as well. The authors took the text, transformed it using a transformer, and then used the resulting token embedding as a class-conditioning in the diffusion model. Each layer of the model's attention layer also pays attention to every text token that the transformer generates as it encodes the text. To increase the text's persuasiveness in terms of image formation from text, the authors, Nicole et al. [26] experimented with Contrastive Language-Image Pre-Training (CLIP)-guided diffusion. The concept behind this is to utilize a second model to improve how well the generated image matches the text. The additional model in this case is CLIP, a program from OpenAI that has been trained to estimate

the similarity of an image and a text. Before generating an image with CLIP-guided diffusion, the model is initially used to denoise the image based on text. They also add a gradient of CLIP's image-sentence similarity to the image, this shifts the initial denoised image in the direction that CLIP predicts will result in a strong image text match and is called as classifier-guided diffusion. The classifier-free advice employed by the authors is another approach and it worked better for them. As the name already implies, no additional model is required. It is an unique approach employed at each level of diffusion to accentuate the text even more. The image is created twice by the model, once with access to the text and once without. Then, using the difference between the diffusion step with text and without text. This difference is used to determine the way to proceed in order to transition from no-text to text. The output of the model without text information is, therefore, strongly projected in the direction of text information if we take the text-less generation and add this difference scaled by a large amount. Although Guided Language to Image Diffusion for Generation and Editing (GLIDE) has approximately four times less parameters than DALL-E and was trained using the same data as DALL-E, it excels in photorealism [26]. The majority of participants in the human evaluation trials clearly favoured GLIDE's generations to DALL- E's fuzzier and messier outputs [26].

2.2 Overview of GAN based Approach for Text to Image Synthesis

Before GANs, Mansimov et al. [24] used a recurrent neural network to create images from text captions. It focused on creating the image in multiple steps, similar to DRAW by Gregor et al. [16]. Reed et al. [34] later demonstrated improved image generation using a generative adversarial network rather than a recurrent variational auto-encoder [45]. Progress was made over the next few years using various methods, which included modifying the generative model architecture. The work done since Mansimov et al. [24] has resulted in appreciable improvements in visual quality.

However, severe defects such as object deformation, incorrect item placement, or an unnatural blending of foreground and background elements can still appear in some examples [32].

Constructing image pixels from text human written descriptions is a complex task. In order to address the challenge, we need to understand how to learn text feature representations that will have visual details and to use these features to generate images that are real enough to trick a person. Image generation using generative adversarial networks (GANs) was introduced by Goodfellow et al. [15] . The GANs consist of a generator and a discriminator that play the minmax game [15]. This minimax game helps them by challenging each other and, thus, training them to be able to generate better images. While at first, the samples are not good and are rejected with confidence by the discriminator, the generator trains itself to be better over time [15].

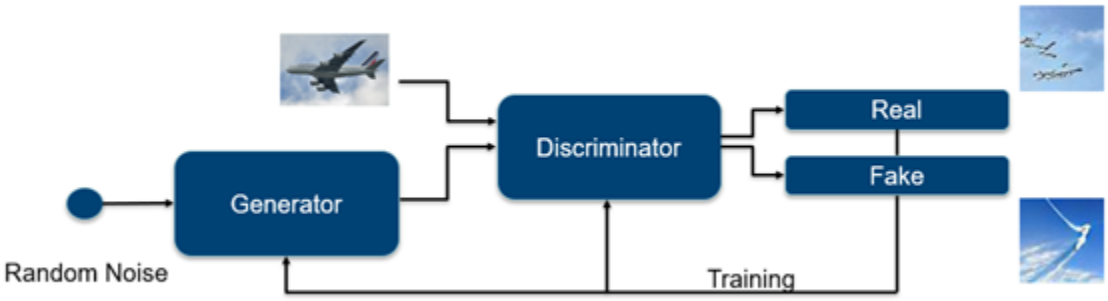

Fig. 2.2.1: GAN Architecture

The Minimax Game [15]:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))] \qquad (1)$$

The generator and the discriminator are competing against each other during the training process. While the generator is attempting to deceive, the discriminator is attempting not to be deceived. When combining both aspects together, a min-max game is placed between the generator(G) and the discriminator(D) with value

function V. While the discriminator is trained, it classifies both the real and fake data from the generator.

x is data representing an image. $p_z(z)$ is the input noise variable, and $G(z)$ is generation of data with random noise z as input. $D(x)$ is trained to identify generated and real instance. It represents the probability that x came from the data rather than generator. The model penalizes itself for misclassifying a real instance as fake, or a fake instance (created by the generator) as real, by maximizing equation 1. The likelihood that the generator correctly classifies the real image is denoted by $log(D(x))$. Maximizing $log(1 - D(G(z))$ would aid in correctly labelling the generated fake image. The generator's loss is then calculated from the discriminator's classification, and it gets rewarded if it successfully fools the discriminator and penalized otherwise. Generated instances serve as negative training examples for the discriminator, which learns to distinguish between fake and real data and penalizes the generator for producing unconvincing results.

In terms of creating synthetic images from real-world images, generative adversarial networks (GANs) have demonstrated promising outcomes [15]. GAN must first be trained to produce high-resolution, photorealistic images from text descriptions, which is an extremely challenging task. In modern GAN models, simply adding more upsampling layers to generate high-resolution (e.g., 256×256) pictures typically leads to training instability and provides meaningless outputs [45].

Gregor et al. [16] presented a Deep Recurrent Attentive Writer (DRAW) neural network design for generating images. The encoder network compresses the real images shown during training, while the decoder network reconstructs images after receiving the codes. These two recurrent neural networks are the foundation of the DRAW architecture. The encoder and decoder are both recurrent networks, which allows them to exchange a series of code samples. Additionally, the encoder has access to the decoder's prior outputs, allowing it to modify the codes it transmits in accordance with the decoder's past behavior. Instead of emitting this distribution all at once, the outputs of the decoder are sequentially added to the distribution that will finally create the data. The input region that the encoder observes and the output

region that the decoder modifies are both constrained by a dynamically updated attention mechanism. MNIST [7] database is a massive collection of handwritten digits. The experiments demonstrated the models ability to outperform MNIST image generation and generate realistic images.

Mansimov et al. [24] introduced the AlignDRAW model, which uses a soft attention method to create a generative model of images from captions. The alignDRAW model used a deterministic laplacian pyramid adversarial network [8] to refine the images produced by the model in a post-processing stage. This approach was built on an extension of the Deep Recurrent Attention Writer (DRAW) [16], which repeatedly creates patches while paying attention to the pertinent description words [24]. The model, which combines an alignment model over words with a recurrent variational autoencoder, was successful in producing pictures that match a given input caption. The model benefited from various factors due to the intensive use of attention processes. In other words, by using the visual attention mechanism, the authors were able to break down the challenge of creating images into a series of steps rather than a single forward pass, and by using attention over words, they were able to gain insight whenever the model was unable to produce a pertinent image. Furthermore, the algorithm produced images with captions that went outside the training set, such as unexpected circumstances that are extremely unlikely to occur in real life.

Reed, Akata, Yan, et al. [34] proposed a text-to-image generation approach that aims to scale up the model to higher resolution images and add more types of text. The model was able to combine a variety of logical visual interpretations of a given text caption [34]. The study demonstrated how style and content could be separated, as well as the posture and backgrounds could be transferred from images to written descriptions [34]. The text characteristics are stored via a hybrid character-level convolutional recurrent neural network and then trained into a deep convolutional generative adversarial network (DC-GAN). The discriminator is taught to differentiate between authentic and fraudulent image-text combinations. The text characteristics are stored via a hybrid character-level convolutional recurrent neural network and then trained into a DC-GAN. Feed-forward inference is carried out by the generator

network and discriminator network based on the text feature. The model is unique in that it is entirely a GAN rather than employing GAN just for post-processing, and can frequently produce aesthetically convincing 64×64 pictures conditioned on text. Prior to encoding the text query using a text encoder, it samples from the noise in the generator. A fully connected layer is utilized to compress the description embedding to a minimal dimension, which is then followed by leaky-ReLU before being concatenated to the noise vector z. Then, inference happens as it normally would in a deconvoluted network. Pass it forward to the generator to create the synthetic image x. Using query text and a noise sample as inputs, the generator performs feed-forward inference to generate images. Multiple layers of stride-2 convolution with spatial batch normalization and leaky ReLU are used in the discriminator. To reduce the description's dimensionality following correction by embedding it in another fully connected layer. Lastly to compute the final score from discriminator, use a 1×1 convolution followed by rectification and a 4×4 convolution. With the help of the findings from the MS-COCO data set, it was shown how generalized the method was for creating images with various objects and changing backgrounds.

Following research suggested the use of several stacked generators to enable text-to-image models to synthesize better quality images [13]. In StackGAN [45], the authors introduced a brand-new stacked generative adversarial network to generate photorealistic images from text descriptions. It greatly enhances the state of the art by breaking down the challenging task of producing high-resolution pictures into smaller, more manageable subproblems [45]. The first stage of StackGAN [45] uses a text conditioning vector and a random noise vector to build a coarse $64\times$ 64 pixel picture. A second generator takes this initial image and the text embedding and generates a 256×256 pixel image. A discriminator is taught to tell the difference between image-text pairings that match and which do not at both phases [13]. The authors present easy-to-use but powerful stacked generative adversarial networks to generate high-resolution images with photorealistic features. It divides the process of generating images from text into two steps [45]. Stage-I GAN creates a low-resolution image by first sketching the basic shape and colours of the item based on the text

description provided and then laying out the backdrop using a random noise vector. Stage-II GAN creates a high-resolution, photo-realistic image by fixing flaws in the low-resolution image from Stage-I and finishing the object's features by reading the text description once more. Zhang et al. [45] further enhanced the architecture (StackGAN++) with an end-to-end framework in which three generators and discriminators are collaboratively trained to simultaneously approximate the multi-scale, conditional, and unconditional image distributions [13]. Instead of using the fixed text embedding produced by a pre-trained text encoder, the authors of AttnGAN [43] used Skip-Thought vectors [21] and StackGAN [45]. The authors also demonstrated the inferiority of conventional text representations like Word2Vec which is a algorithm that uses a neural network model to find connections between words in a large text corpus. The latent variable is sampled at random from a Gaussian distribution, where the text embedding determines the mean and covariance matrix. Many of the text-to-image methods that came after this one utilized this method.

OpenAI launched a revolutionary deep neural network that learns visual ideas via natural language supervision. Two encoders make up CLIP (Contrastive Language-Image Pretraining), one for texts and one for images. It is sufficient for each image to verify whether the text description "a photo" or "photo of X" is more likely to be matched with the image of an item X in an image collection [14]. The authors Galatolo et al. [14] introduced a CLIP-based framework called CLIP-guided Generative Latent Space Search (CLIP-GLaSS), which is used to produce the best images matching a target caption. After being explored by a genetic algorithm, this ideal image (or text) is generated by a generative network. Models like StyleGAN2 and generator GPT2 have been used in preliminary experiments of the proposed CLIP-guided generative latent space search (CLIP-GLaSS) for the text-to-image tasks. The results show that the suggested framework has a lot of potential in terms of the quality and usefulness of the picture or text that comes out, which calls for more comparative research.

2.3 Overview of Diffusion Model Approach for Text to Image Synthesis

Diffusion models are a new category of cutting-edge generative models that produce a variety of high-resolution pictures. There are currently a variety of models based on diffusion. The initial denoising diffusion technique was developed by Sohl-Dickstein et al [39]. They create a technique that simultaneously provides adaptability and manageability. The basic idea, which comes from non-equilibrium statistical physics, is to use an iterative forward diffusion process to systematically and gradually get rid of structure in a data distribution. Then, they discovered a reverse diffusion process that recovers data structure, resulting in a highly flexible and tractable generative data model. This method permits the quick learning, sampling, and evaluation of probabilities in models with hundreds of layers or time steps, as well as the computation of conditional and posterior probabilities under the learned model. The result is an algorithm that can learn to fit any data distribution, is easy to train, test, and create samples from, and makes it easy to change conditional and posterior distributions [39].

In diffusion models, a distribution is sampled by reversing a slow noise process. Each timestep t corresponds to a different level of noise, x_0 is signal and x_t is a signal mixed with noise, with the signal-to-noise ratio determined by the timestep t [10]. Sohl-Dickstein et al. [39] suggested diffusion probabilistic models that learn to reverse a laborious, multi-step noise process to suit a data distribution [27]. Ho et al. [19] developed a novel explicit relationship between diffusion models and denoising score matching, which resulted in a reduced, weighted variational bound objective for diffusion models. The diffusion model creates a Markov chain of latent variables x_1,... x_T by gradually adding Gaussian noise $q(x_0)$ to a sample from the data distribution x_0 [26].

$$q\left(x_t \mid x_{t-1}\right) := \mathcal{N}\left(x_t; \sqrt{\alpha_t} x_{t-1}, \left(1-\alpha_t\right) \mathcal{I}\right) \tag{2}$$

The posterior $q(x_{t-1}|x_t)$ is clearly defined if the magnitude $1–\alpha_1..\alpha_T$ of the noise supplied at each step is large enough [26]. x_t is well approximated by $\mathcal{N}$. These qualities propose learning $p(x_{t-1}|x_t)$ to approximate posterior by the below given equation 3 [26].

$$p_\theta\left(x_{t-1} \mid x_t\right) := \mathcal{N}\left(\mu_\theta\left(x_t\right), \Sigma_\theta\left(x_t\right)\right) \tag{3}$$

It is possible to use it to generate samples of the form $x_0 \sim p_\theta(x_0)$ by beginning with Gaussian noise of the form $x_t \sim \mathcal{N}(0, I)$ and then progressively lowering the noise in the form of a series of steps $x_{t-1}, x_{t-2}, ..., x_0$ [26]. To create samples $x_t \sim q(x_t|x_0)$ by adding Gaussian noise to x_0 in order to compute this surrogate goal, and then train a model to anticipate the additional noise using a common mean-squared error loss.

Diffusion models seem to have good inductive biases for image data. The best results came from training on a weighted variational bound that was made based on a new connection between diffusion probabilistic models and denoising score matching with Langevin dynamics. The models naturally allowed a progressive lossy decompression scheme that can be thought of as a generalization of autoregressive decoding. According to Ho et al. [19], the basic mean-squared error goal performs better in reality than the actual variational lower bound determined by interpreting the denoising diffusion model as a VAE [9]. Denoising Diffusion Probabilistic Models (DDPM) are an attractive choice for generative modeling since they combine log-likelihoods, high-quality samples, and reasonably fast sampling with a well-grounded, stationary training objective [27]. The authors, Nichol et al. [27], found that DDPMs can match the sample quality of GANs while achieving much better mode coverage as measured by recall. Denoising diffusion implicit models (DDIMs), introduced by Song et al. [40], are an implicit generative model trained with denoising auto-encoding and score matching objectives. It can generate high-quality samples much more efficiently than existing DDPMs. The non-markovian forward process shown here seems to suggest continuous forward processes that are not gaussian, which can not be done in the original diffusion framework.

Nichol et al. [26] introduced Guided Language to Image Diffusion for Generation and Editing (GLIDE) with classifier-free guidance that is capable of generalizing to a wide variety of text prompts. The model can generate realistic shadows and reflections, as well as high-quality textures. It is also capable of producing illustrations in various styles, such as the style of an artist or painting. GLIDE outperforms classifier-free guidance when it comes to matching the prompt. The model achieves competitive FID on MS-COCO without ever explicitly training on this dataset. GLIDE was trained with roughly the same training computation as DALL-E but with a much smaller model. Diffusion models have been improved in various recent papers. OpenAI, Nvidia, and Google have successfully trained large-scale models, which have received considerable interest. GLIDE [26], DALLE-2 [31], and Imagen [36] are complete open-source tools that are examples of designs that are based on diffusion models [20].

2.4 Comparison of Generative Models

Most of the previous research has mostly concentrated on analyzing generative models of single or few object scenarios or facial features. Comparatively less work has been done to evaluate how well the generative model can generate scenes with a higher level of complexity [34]. The development of automatic metrics that are consistent with human judgment and that enable rankings of models that are accurate as well as meaningful is a problem that must be met [9]. Frolov et al. [13] presented a review of the generative adversarial methods with emphasis on text-to-image synthesis. These new areas of research includes the creation of improved data sets and evaluation metrices to the possibility of advancements in architectural design and model training. In addition to this, they investigated the most frequently assessment methods in order to evaluate image quality and the image-text alignment. The use of automated measures such as the Inception score [1], Frechet inception distance [3], and SOA [18] has made the process of evaluating models much more straightforward.

Crowson et al. [5] employed human subjects who were asked to assign a score

on a scale ranging from one (low) to five (high) to indicate how well text and image combinations are aligned. They were told to rate higher-quality pictures that didn't match the prompt lower than lower-quality pictures that did align with the text caption. By calculating the average score per text caption used for each model, they determined that, the participants believe that the generations that utilized their model are more aligned with the text captions.

Using SOA introduced by Hinz et al. [18], evaluation of several state-of-the-art approaches demonstrated that no method used at the moment can provide accurate background elements for the 80 classes in the MS COCO data set. Some models are adequate for common objects but they all fail for unusual objects or those without an easily recognizable appearance. Using SOA to evaluate text-to-image models offers more precise information about how well they perform for different object classes or image captions and is aligned with human evaluation.

2.5 Data sets

Deep learning algorithms require large, high-quality data sets to be successful. Oxford-102 Flowers [28] and CUB-200 Birds [41] are two of the most popular text-to-image data sets. They are relatively simple to use and contain approximately 10,000 images with five human-generated captions per image. Both the CUB-200 Birds and the Oxford-102 Flowers data sets are also referred to as "single item data sets" because they contain only one item per image [18]. CelebA-HQ [23] is another data set with a single object. The captions or code to replicate for the CelebA-HQ data set are not open-sourced yet [18]. MS COCO, on the other hand, contains 123,000 images, where each image may contain multiple objects, and there are associated single-sentence captions [13]. Over time, generative models have progressed from generating single-object picture data sets like the CUB-200 data sets to multi-object image data sets like the MS COCO. The more advanced models are utilizing data from the internet which gives them an upper hand on previously trained models. The type of output expected from the generative model depends on the data we used to

train it. A model trained on the data set for CUB-200 birds won't be able to generate images of faces. It's not necessary for the input caption used for the generation to match with the training data set. Although the model is expected to manage the variety of text input and generate images, it can not generate an object that it has not been trained on.

CHAPTER 3

Methodology

3.1 Motivation

Within the last five years or so, the area of text-to-image generation has evolved considerably and quickly in terms of the complexity of the data sets employed, the resolution of the generated images, and their quality. This technology marks a significant shift since it eliminates the need for technical work and manpower in the creation of images. Instead, they look for original thinking and curatorial judgement. The long-term implications are hard to predict, but these algorithms point to a new, democratic way of expressing ideas that may lead to a huge increase in the number of pictures made by humans, just like the camera and digital camera did [11].

For the majority of image generating tasks, GANs have been the most advanced technique. AlignDRAW, a variational recurrent autoencoder built upon DRAW proposed by Gregor et al. [16], is not entirely based on the generative adversarial network approach, but it uses GAN to only sharpen the generated image. We now have various versions of GAN available with much improvement in performance for image generation and the quality of the generated image. GANs are sometimes challenging to train because they collapse in the absence of precisely chosen hyperparameters [9]. Later, the diffusion models were introduced, which turned out to be better than GANs at creating images from text [13, 9]. However, changes were made to the GAN model, which led to a variation of GAN outperforming diffusion models by making better-looking images than earlier, less flexible methods [5]. This field has also developed quantitative evaluation metrics that are used to evaluate the quality of text-to-image

synthesis models. Frechet inception distance [3] and Inception Score [1] are examples of metrices used to evaluate generative models. While some papers have not employed all possible metrics and others haven't employed any quantitative metrics at all, there are other metrics like Semantic Object Accuracy (SOA) that evaluate the model while also taking the text used to generate it into account. In this thesis, we have set up a way to evaluate these two different generative models. This evaluation takes into account the quality of the image and how well it fits with the meaning of the text.

3.2 Proposed Method

After reviewing the preceding chapters about generative models, in this section we now seek to investigate the modification of the GAN model which is VQGAN - CLIP and the diffusion model GLIDE that we are going to use in our comparative analysis. We will also shine some light on text encoders, and the data set used. The methodology for the evaluation comprises of three steps: we start with the implementation of both the aforementioned models, then progress onto the generation of image data sets utilising captions from the MS COCO data set. We then leverage these generated images to assess the model's capacity to produce high-quality images and analyze these images against the captions from the aforementioned data set.

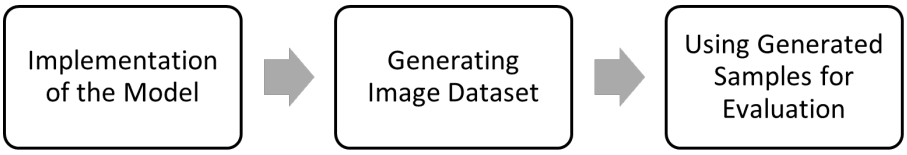

Fig. 3.2.1: Process Flowchart

We use the generated samples for the evaluation of both the models using metrics like Inception Score, FID and SOA which will be discussed further in this chapter.

3.2.1 VQGAN

VQGAN is a GAN modification that produces images with higher visual quality than diffusion models [5]. It consists of two machine learning models combined and was introduced by Esser et al. [10]. Fig. 3.2.3 shows the graph integrating GAN, VQ-VAE, and transformers to construct the VQ-GAN model. The approach aims to understand both the long-range dependencies between the phrases in a sentence and the visual elements of an image. They have employed transformers to teach dependencies, and a GAN to learn the visual components. The transformers scale quadratically and calculate the pairwise inner product between each pair of tokens because of the attention mechanism [25]. It uses this technique to discover long-term relationships between tokens or visual components.

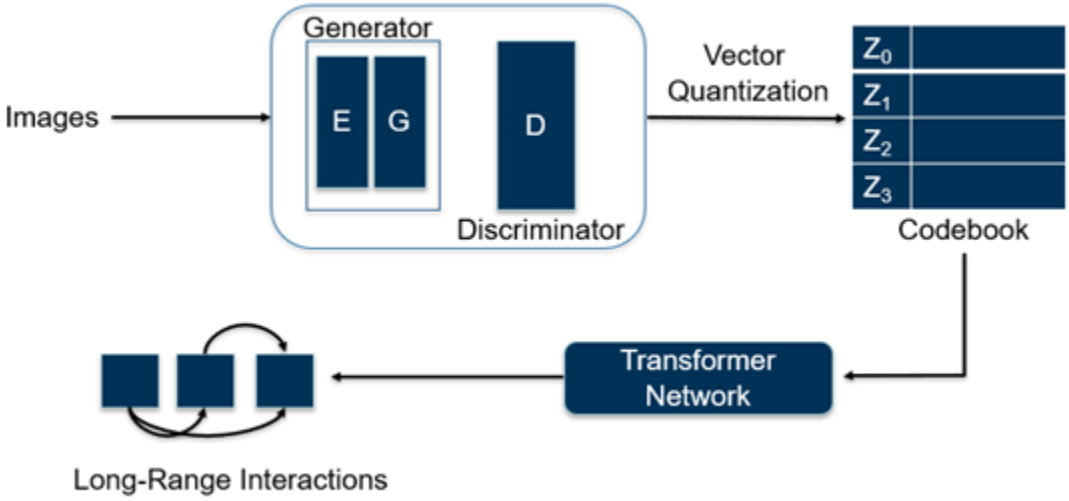

Fig. 3.2.2: VQGAN Architecture [10]

To encode the feature map of the visual portions of the images, the feature map of the image data is first directly supplied to a GAN. Then, this image data is "vector quantized," a type of signal processing that organizes vector groups into clusters that may be accessed by a representative vector designating the centroid and is referred to as a "codeword" (Algorithm. 3.2.1). The vector quantized data is encoded and stored as a codebook, or dictionary of codes, as depicted in Fig. 3.2.2. The image data is

first represented by the codebook, which serves as an intermediary representation, and is then entered as a sequence into a transformer [25]. After that, the transformer is trained to simulate the composition of these encoded sequences as high-resolution images. The generator follows an encoder-decoder architecture. This setup is similar to autoencoders, where the goal is to have a decoder properly reconstruct the input [25]. If the reconstruction is perfect, then the encoder has found a good way to represent the data.

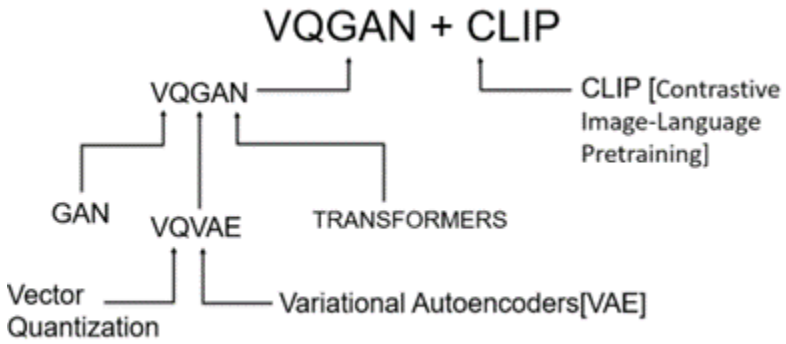

Fig. 3.2.3: VQGAN Architecture [10]

3.2.2 CLIP

The CLIP model was developed to evaluate how well a caption fits with an image when compared to the other captions in the collection. Since CLIP is capable of zero-shot learning, it can function successfully even with untested data [30]. CLIP is particularly good at determining whether a picture and a brief bit of text go together or not. It is not limited to the classes in the data set but knows mostly all english terms, allowing it to formulate ImageNet classes into prompts that include more language than simply the classes. It can model a concept based on its meaning, expanding its capabilities far beyond a limited set of classes and demonstrating excellent zero-shot performance on previously unseen data sets [30]. As a result, CLIP never loses or forgets additional components of the image that were not caught in class, such as the wall or sky in the background, because it is never constrained dur-

ing training to compress an image to a single notion or word [17]. The model takes a set of images and a collection of textual descriptions that go with them. Both the text and the images are encoded using either a ResNet or a Transformer [17]. The encoder then computes an image vector for each image in the batch, with the first image corresponding to vector I_1, and the subsequent images to vector I_n. The same is true for the textual description, with the first textual sequence being encoded by one vector T_1 and the subsequent descriptions by T_n. Between each of these image and text vectors, n squared similarities are calculated. As a result, since I_1 fits with T_1, I_2 fits with T_2, and so on, CLIP should maximize the diagonal elements. The off-diagonal elements should be reduced in a contrastive manner since we believe it is highly improbable that an image I_1 would fit with any random description other than its own. CLIP's can predict similarities between images and textual descriptions, as well as contrastive training. When these image and text encoders are transformed, a lot of computations can run in parallel [17]. CLIP has been trained to forecast high similarity for suitable image-text pairs and low similarity for random ones, and it is ready to be applied to a variety of tasks, including image recognition.

3.2.3 VQGAN - CLIP

When combined, VQGAN - CLIP (Fig.3.2.4) develops the model that can be used to produce images from text. CLIP can assess the quality of generated images compared against a user input caption, and the output scores can be used as weights to guide the learning of the VQGAN to match the subject matter more accurately through recursive iteration [25].

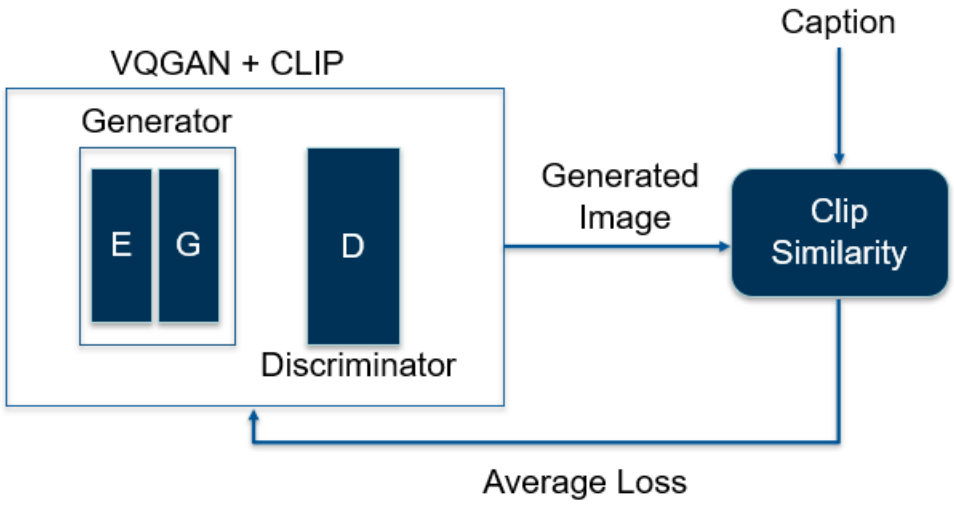

Fig. 3.2.4: VQGAN - CLIP Architecture [5]

Algorithm 3.2.1 Algorithm of the VQGAN - CLIP Model [5]

Input: image i, text t, number of steps S
Initialize: encoder(e), decoder(d), generator(N), discriminator(D), Vector Quantization Module(VQ)
$i \rightarrow$ N(e,d) $\rightarrow$g(i)
VQ(g(i))$\rightarrow$ *codebook*
codebook $\rightarrow transformer$(to learn interaction between visual features)
codebook $\rightarrow$ *decoder* $\rightarrow$ *reconstructed*(i)
update reconstruction loss
update VQ loss
reconstructed(i) $\rightarrow D \rightarrow$ Real or Fake
Update D loss
reconstructed(i) $\rightarrow CLIP$
$t \rightarrow CLIP$(similarity score) $\rightarrow$ average loss
update N(e,d) with average loss
Repeat for multiple epochs (S) until the models converge

The VQGAN initially creates a random noise image that is vector quantized and encoded in a codebook to create these images. The codebook is then used as input to a transformer that produces the new image from the encoded signals. The output is then used to assess the image's accuracy to the input prompt using CLIP, and the scoring is then sent back to the VQGAN to update the image generation model to reflect the prompt more closely.

3.2.4 GLIDE

GLIDE is trained on a 3.5 billion parameter diffusion model that uses a text encoder to condition on natural language descriptions [26]. Using human and automated evaluations, the authors found that using classifier-free guidance yields higher quality images [26]. The model can render a wide variety of text prompts, but it can have difficulty producing realistic images for complex prompts [26]. Nichol et al. [26] trained a 3.5 billion parameter text-conditional diffusion model at 64×64 resolution for primary experiments, and a text-conditional upsampling diffusion model with 1.5 billion parameters and increased resolution to 256×256. According to Dhariwal and Nichol [9], samples from class-conditional diffusion models can frequently be enhanced with classifier guidance. The CLIP guidance model has been trained to use classifier guidance on GLIDE [26]. For fast sampling, the base model employs 100 diffusion steps, followed by another 27 for upsampling. GLIDE leverages its own implementation of the diffusion model to perform its image generation. Forward diffusion and reverse diffusion are the two stages of its operation. During the backward diffusion process, the model learns to reverse the effect of the added noise on the images and guide the generated image towards its original form [26].

Algorithm 3.2.2 Algorithm of the GLIDE Model [26]

Input: noised image $n(i)$, text t, number of steps S
Output: image i
initial image = $n(i)$
objects(o), attributes(a) $\leftarrow transformer$(t)
for each-step in range(1,S):
  $predictions(pred) \leftarrow DiffusionStep$(i)
  $diff \leftarrow CLIP(pred, o, a)$
  n(i)$\leftarrow update(i, diff)$
return i

The model starts with a noise image and a text description as input. GLIDE offers some level of control over the result of the picture production process by parsing the text input prompts. This is accomplished by training the transformer model on a sizable data set made up of pictures and the descriptions that go with them [38]. The text is first converted into a string of tokens, which are then used as conditions.

The transformer model is then fed these tokens. There are then two uses for the transformer's output. The class embedding for the diffusion model is first replaced with the final token embedding obtained after conversion. Embedding is translating something from high dimensional space to low dimensional space. The last layer of the token embeddings, which is a series of feature vectors, is then independently projected to each attention layer's dimensions in the model and concatenated to each attention layer's context [38]. This information is then used to guide the generation of the image, by iteratively updating the pixels in the noise image based on the desired properties specified in the text description (Algorithm 3.2.2). At each iteration, the model uses a set of steps to process the image and make predictions about the desired properties. The model then compares these predictions to the desired properties specified in the text description, and adjusts the pixels in the image accordingly. This process is repeated until the generated image is deemed photorealistic enough, or until a maximum number of iterations has been reached.

This enables the model to create an image from novel combinations of related text tokens in a distinctive and photorealistic way by using its learned understanding of the input words and their associated images. This text-encoding transformer has over 1.2 billion parameters and employs 24 building blocks, each with a size of 2048 [38]. The up-sampler diffusion model, which includes 384 base channels and a text encoder with a size of 1024 and has around 1.5 billion parameters.

3.3 Dataset

We are using the MS COCO data set for our evaluation. We choose all image captions from the MS COCO validation set that specifically refer to one of the 80 major object categories (such as "human," "dog," "car," etc.) [18]. The data set consists of more than 80,000 images. The data set is created by compiling typical complicated scenes of everyday items. There are at least five captions for each image. We are selecting one caption for each image. It is crucial for our experiment that the chosen caption includes the object class in the sentence.

3.4 Evaluation Metrics

It is crucial to have access to automatic evaluation criteria that fairly compare performances and measure improvement. Evaluating generated images is particularly difficult because there are so many characteristics that resemble a good image, such as visual realism and diversity [13]. A good text-to-image model, however, encompasses more than just producing realistic visuals. The semantic alignment of generated images and text descriptions is a crucial additional consideration. The training data distribution should be accurately portrayed in the images created from text descriptions. We are using the matrices , Inception Score and Fréchet Inception Distance. Barratt et al. [1] demonstrated Inception score provides beneficial guidance when evaluating and comparing models. Heusel et al. [17] introduced the FID score which more accurately represents how similar generated images are to real images. We will discuss about another metric which we have used i.e., the semantic object accuracy (SOA). Unlike the majority of the existing evaluation measures, this metric focuses on specific elements and components within an image and also takes the caption into account when rating a picture. Often explicitly or indirectly, image captions describe the objects that can be observed in an image, such as an "A person using a cell phone" is the description of the photo should show a person and a cell phone together [18]. The evaluation metrices that we have used for our comparative analysis and experimentation will be discussed.

3.4.1 Inception Score (IS)

An algorithm for measuring the effectiveness of image generative models is called the Inception Score. This metric is shown to correlate well with human scoring in terms of the realism of generated images. It uses an Inception v3 network that has already been trained on ImageNet and figures out statistics about the network's results when applied to images that were made by a computer [12].

$$IS(G) = exp(E_{x \sim pg} D_{KL}(p(y|x) \mid\mid p(y)) \tag{4}$$

The images generated should contain clear objects, i.e., the images should be sharp rather than blurry, or $p(y|x)$ should have low value. In other words, the Inception Network should be highly confident that there is a single object in the image [1]. The generative algorithm should output a high diversity of images from all the different classes in ImageNet, or $p(y)$ should have high value. The KL-divergence is statistical distance that quantifies the differences between two probability distributions. If a generative model has both of the aforementioned characteristics, we would expect a large KL-divergence between the distributions of $p(y)$ and $p(y|x)$, which would lead to a large IS [1].

3.4.2 Frechet inception distance (FID)

FID is a statistic that measures the distance between the feature vectors of real and fake images generated by the model. Heusel et al. [17] first presented it in 2017. A higher quality and closer resemblance to real images are indicated by a lower FID score for the generator. The foundation of FID is an image's feature vector. If FID is your performance metric, attempt to keep it as low as possible, but having a very low score can also mean that the images are very similar and cause a lack of diversity in the data set we are trying to generate. Fréchet inception distance (FID) is a measurement used to evaluate the quality of the generated images.

$$d^2 = ||\mu_1 – \mu_2||^2 + Tr(C_1 + C_2 – 2 * sqrt(C_1 * C_2)) \tag{5}$$

The score is denoted by the symbol d^2, indicating that it is a distance in square units. The terms μ_1 and μ_2 designate the feature-wise means of the actual and artificial images, respectively. These vectors have 2,048 elements each, each representing the mean feature that was seen across the images [3]. The covariance matrices, C_1 and C_2, represent the real and produced feature vectors, respectively. The expression $||\mu_1 - \mu_2||^2$ denotes the total squared difference between the two mean vectors. The linear algebra operation Tr stands for the sum of the elements in the square matrix's major diagonal [3]. An Inception v3 model that has already been trained is loaded

before calculating the FID score. The output is taken as the activations from the final pooling layer, a global spatial pooling layer, after the output layer of the model has been removed [3]. Each image is projected to have 2,048 activation characteristics since this output layer has 2,048 activations. This is referred to as the image's coding vector or feature vector. Then, in order to demonstrate how genuine images are represented, a 2,048 feature vector is projected for a selection of real photos from the issue domain. Then, feature vectors for newly created photos can be computed. Two collections totaling 2,048 feature vectors for both real and created photos is the result [3].

3.4.3 Semantic Object Accuracy (SOA)

Most evaluation metrics do not take the image caption into account and do not evaluate individual areas or objects within an image. To address this, Hinz et al. [18] introduced a novel evaluation metric based on a pre-trained object detection network. This metric is called Semantic Object Accuracy (SOA), which measures directly whether objects mentioned in the caption are recognizable in an image as well as whether the image contains them. Similar techniques have been employed in several earlier publications to assess the quality of the generated images [18]. But since the contents of the caption aren't taken into account, any detection with a high level of confidence is still good, even if the object being detected doesn't make sense in the context of the caption [18]. Images are generated for each of the 80 objects in the MS COCO data set by analyzing the captions in the validation set. Captions are filtered for keywords that are related to the available labels for objects. It uses all captions that imply the existence of each object in the data set to produce three images for each object [18]. The YOLOv3 network [33], pre-trained on the MS COCO data set, is then applied to each of the generated photos to see if it can identify the specified object [18]. They presented the recall as a class average (SOA-C) (equation 6), which represents the average number of images per class in which the YOLOv3 detects the given object (equation 7) and image average (SOA-I), which represents the average number of images in which the required object was detected [18].

$$\text{SOA} - \text{C} = \frac{1}{|C|} \sum_{c \in C} \frac{1}{|I_c|} \sum_{i_e \in I_e} \text{YOLOv3}\left(i_c\right) \tag{6}$$

$$\text{SOA} - \text{I} = \frac{1}{\sum_{c \in C} |I_c|} \sum_{c \in C} \sum_{i_c \in I_c} \text{YOLOv3}\left(i_c\right) \tag{7}$$

$i_c \in I_c$ are images that are supposed to contain an object of class c.

$$\text{YOLOv3}(i_c): \begin{cases} 1 \text{ if YOLOv3 detected an object from class c} \\ 0 \text{ otherwise} \end{cases}$$

In order to determine whether generated photos contain objects that are specifically referenced in the image caption, we employ a pre-trained object detector. An object detector for the provided data set is all that is required for this. There are many of good pre-trained object detectors accessible for the MS COCO data set. To calculate the semantic object accuracy, we obtain all captions that expressly refer to each of the 80 foreground objects that are given a label in the MS COCO data set [18]. For each caption, we generate images using both the generative models. Use the object detector to determine whether the generated images has the object that it ought to contain. This object detector is a pre-trained YOLOv3. Calculate the frequency with which a particular object was found in the images generated for each label. The SOA fixes issues like considering the image caption, analyzes images using an object detector that has already been trained on the same domain. Additionally, it takes into account the foreground objects of the images and avoids overfitting the model during training [18].

CHAPTER 4

Experiments and Results

4.1 Implementation and Tools

In this section, we will go through the tools that we utilized to carry out the research study. Here are the development tools, followed by the implementation environment.

Development Tools:

- Platforms: Google Collaboratory- NVIDIA Tesla K80, Jupyter Notebooks, Anaconda
- Programming Language: Python 3.7
- Libraries: Sklearn, OpenCV, Scipy, Numpy, Pandas, Keras, pytorch

Implementation Environment:

- Operation system: 64-bit Windows 10
- System type: x64 based processor
- CPU: Intel Core i7-8565U CPU
- RAM: 16 GB
- GPU: NVIDIA GeForce RTX 2080

4.2 Results and Discussion

4.2.1 Inception Score Results

We run experiments using the samples created from the two models, which are the VQGAN - CLIP and the GLIDE. For our analysis, 90 images are generated for each class from the MS COCO data set. We are creating three images from each caption. We are using a total of 30 captions for each category. We also combine all the images and then compute the overall Inception score.

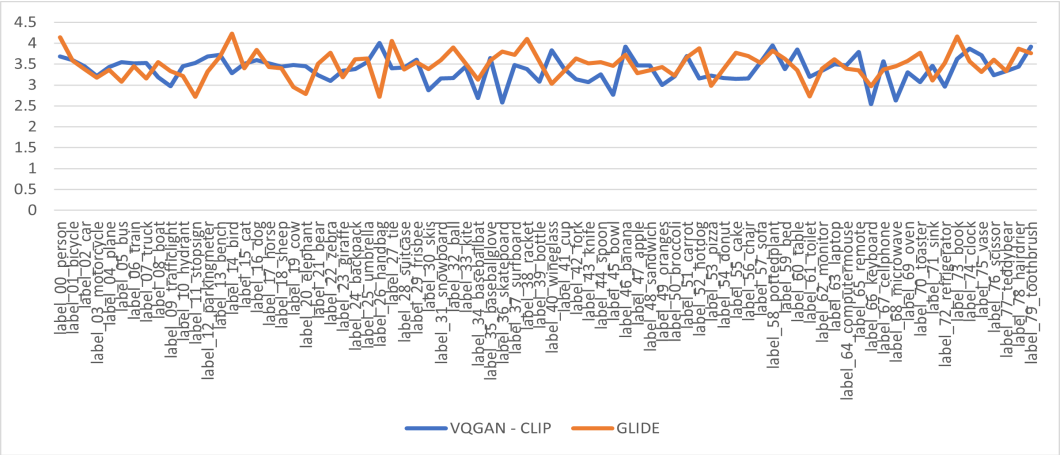

Fig. 4.2.1: Inception Scores for VQGAN - CLIP and GLIDE

Although there has been significant development in recent years in producing images from captions, the process of creating images of complicated scenes with several items that may interact with one another is still highly challenging. We utilize the model's output, to calculate the IS. The overall entropy over the classification outputs for all images should be large, indicating that the generated images contain objects of different classes, while the values for each of the generated images should be small, indicating that the network is confident that there is one specific object in the image. According to our results, the classes "bird", "book", "person" and "racket" have the highest inception score for the GLIDE model relative to other class objects. While the object that the images of particular class is supposed to contain is not visible in the sample images generated, the score seems good for the data set generated for the class "person" because the inception model can detect other objects in the im-

ages. For the class "book," the models are generating images that depict the objects stated in the text, align with the caption, and offer variety to the data set. The VQGAN - CLIP model scored lower than GLIDE because the items were easier to recognize and identify using the inception model. For the "bird" and "racket" classes, the model generates more distinct objects, and the generated image focuses on one object, which leads to a higher IS for GLIDE. Because the images created are blurry, VQGAN - CLIP receives relatively lower scores for the same classes. The images produced by GLIDE are unquestionably more visually appealing. For VQGAN - CLIP, "handbags", "potted plants", "elephant", and "bananas" are the highest. The caption type also plays a significant role in the image generation process. For example, a "bird" with "ground" as the background is relatively easier to create than a "cat" in a "handbag".

Original Image VQGAN - CLIP GLIDE

Fig. 4.2.2: Set of books sitting next to a small black clock.

Fig. 4.2.3: A bird sitting on the ground.

Fig. 4.2.4: A cat sleeping in a red handbag.

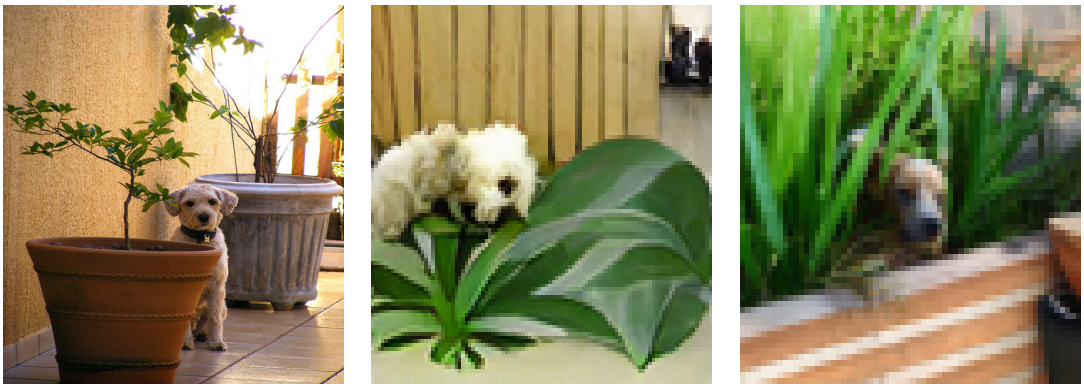

Fig. 4.2.5: A dog is sitting between two large potted plants.

The lowest scores for VQGAN - CLIP are, for classes, “keyboard”, “skateboard”, “microwave”, and “baseballbat”. We observe that for all these objects the caption we are using has more than one uncommon objects. For example, for “mircowave”, the captions, “A microwave on top of a fridge in a kitchen.” or “Microwave on top of a small refrigerator with shelves above”. The model tries to generate one object at a time, resulting in generation of distorted images. These distorted images will have low inception scores as the models could not generate a coherent representation from these features and instead distributed them throughout the image. We can only see the texture and pattern generated for most of them, and not distinct objects. The lack of image text pairs while training of models with both objects together can be one of the reason that the model is unable to generate the image.

For the GLIDE model, lowest-scoring object classes are “elephant”, “zebra”, “stop sign”, and “toilet”. Although most of images generated contain objects the score is

relatively lower than other classes. The score for these classes is also low because most of them are single object images and the object detection model used by IS only detects one object and scores accordingly.

Model Name	IS Score
VQGAN - CLIP	12.86
GLIDE	19.26

Table 4.2.1: Overall IS Scores

The above Table. 4.2.1 gives the overall IS Score for both models. The GLIDE model has a higher overall IS for the images, which tells us it is more effective in producing images. The IS can be substantially greater for simpler data sets. Additionally, we see that the score rises as the size of the data set increases.

4.2.2 FID Results

The FID represents the distance between two images. It is used to calculate the distance between the true image and the generated images. The FID still has the issue that the image statistics are derived using a network that was previously trained on ImageNet [6], which is an extensive image database but may not be a representative data set, for more complicated data sets.

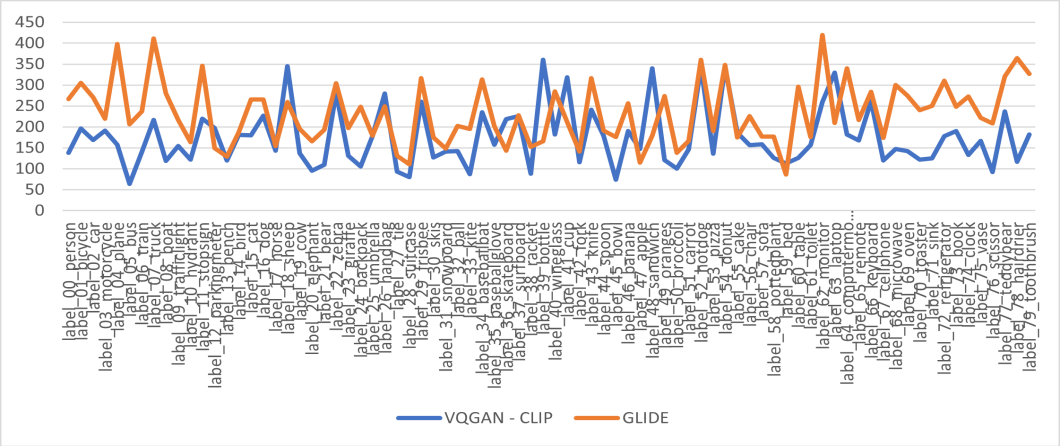

Fig. 4.2.6: FID Scores for VQGAN - CLIP and GLIDE

Further discussion will be on the visuals for both high and low peaks in the graph presented above. Beginning with the items with the highest distance between the original image and the generated image for the VQGAN - CLIP model, the bottle and teddy bear image generated by the VQGAN - CLIP model contains bottles but was unable to recreate the teddy bear as in the original image. The image distribution created by generator will be random every time. The generator focuses on generation of image using caption given to it and tries to recreate from what it has learned during training. The models are trained on various images other than the original image and that is why the generated image has high FID score. We can observe that VQGAN - CLIP is recreating the donuts , but they look different than the original image for the same reason. Next is "sheep", where the VQGAN - CLIP model can only generate the fur texture of the sheep , but the sheep is not visible in the picture. The reason being that model focused on the words "hairy sheep" from the caption and the image distribution created only contains that portion of the caption. While the GLIDE was able to reconstruct the sheep but not with the same texture of fur as the original image. For "sandwich", both the models the did not take into account the second part of the sentence, i.e., "a bowl of soup," for the generation of the image. For the FID scores, we are generating single zero shot images for captions but if we try again the image distribution may contain the "bowl of soup" which is why for the IS and SOA score we are using the same caption to generate 3 images and take the possibility of model generating the other objects into consideration.

Ground Truth VQGAN - CLIP Generated GLIDE Generated

Fig. 4.2.7: A brown teddy bear sitting next to bottles of person care items.

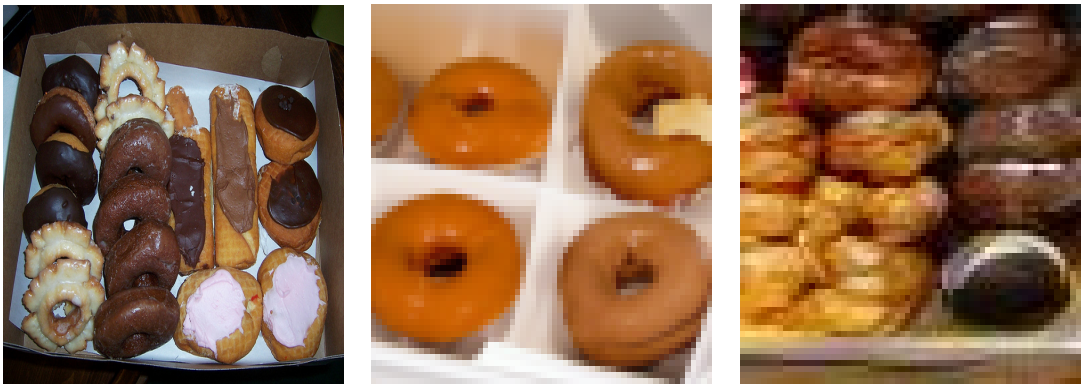

Fig. 4.2.8: A variety of donuts and pastries in a box.

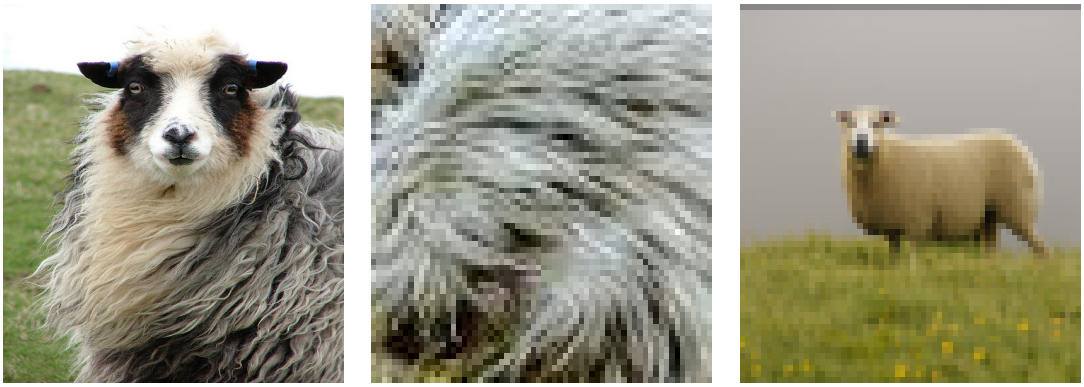

Fig. 4.2.9: A large hairy sheep standing on a lush green field.

Fig. 4.2.10: A table topped with a couple of sandwiches and a bowl of soup.

Even though the images created by GLIDE appear better, they are very different from the original image, giving us variety in the data set. The below-given samples are low FID scores for the VQGAN model. We notice that the "monitor" and "hair dryer" in the samples are created. The GLIDE model can create a similar background

for “trucks” and “airplanes” but not the object mentioned. As we learned earlier about the training process of generative models they try to create the objects first and then work on conditioning, here the model focuses on creation of background as it is mentioned in the caption and then tries to add the other objects mentioned. The model may implicitly place more weight on certain parts of the text description if they are more important for generating the desired image. It appears that VQGAN - CLIP only creates a screen-like structure for the “monitor” class and nothing else. On the other hand, GLIDE recreates a proper desk and monitor and also creates additional objects which belong to the desk as the model has learned that from the training data set for either monitor or keyboard or desk and if all of them are together. Both models cannot create the “hair dryer” and “dog” mentioned in the caption. The models are trying to recreate the objects which is tricking the CLIP to believe similarity to caption but the image is not visually correct. The image generated for the “truck” by VQGAN - CLIP also does not contain the object mentioned but is similar to the overall image, while the GLIDE created a normal road rather than a steep dirt road.

Ground Truth VQGAN - CLIP Generated GLIDE Generated

Fig. 4.2.11: A desk with a computer monitor and a keyboard.

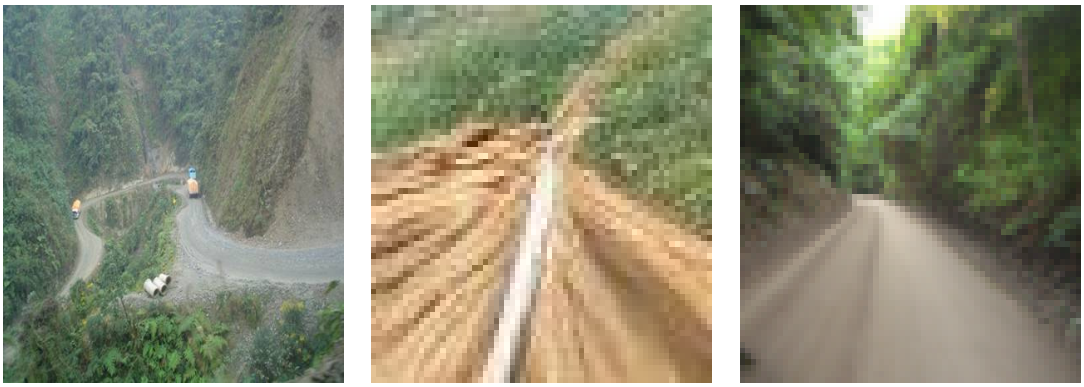

Fig. 4.2.12: Trucks driving down a steep narrow dirt road.

Fig. 4.2.13: A big airplane flying in the big blue sky.

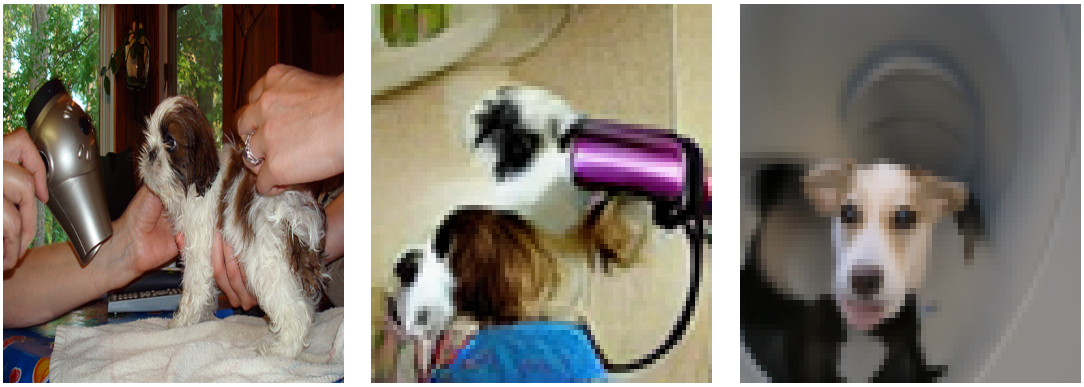

Fig. 4.2.14: Two people are using a hair drier on a small dog.

The sample images in Figures 4.2.15 , 4.2.16, 4.2.17 have low FID scores for VQGAN - CLIP, and the bus is recreated very similarly to the original image. The other class “bowl” is easy to recreate as it contains only one object, and the ‘spoon’ mentioned in the caption is not visible in either of the images created.

Ground Truth VQGAN - CLIP Generated GLIDE Generated

Fig. 4.2.15: A yellow school bus parked in a parking lot.

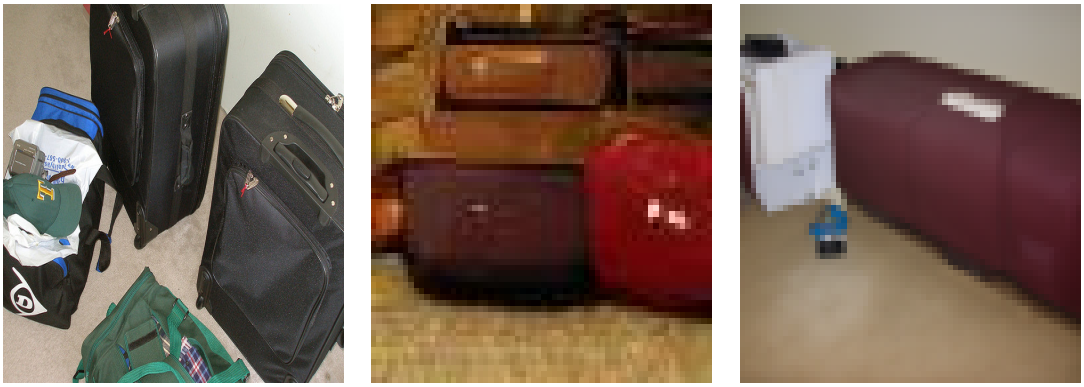

Fig. 4.2.16: Some suitcases and a hat laying on a carpeted floor.

Fig. 4.2.17: A bowl of veggie soup with a spoon in it.

These below given samples of bed (Fig.4.2.18) and apple (Fig. 4.2.19) are highly similar and hence have low scores. It is also because the images contain single objects that are easier to recreate and the objects are not interacting. For "bench"

(Fig.4.2.20) the scores are relatively lower than other objects. Overall, the GLIDE model outperforms the VQGAN - CLIP model at scene recreation.

There will be differences between the original image and the generated images, even if the model is performing well. This is because the generated images are typically based on a noise image and a text description, rather than being an exact copy of the original image. As such, the generated images may contain some variations or deviations from the original image. Comparing the original image to the generated images can provide valuable insights into the performance of the image generation model, but it is important to consider the limitations of the generated images as well.

Ground Truth VQGAN - CLIP Generated GLIDE Generated

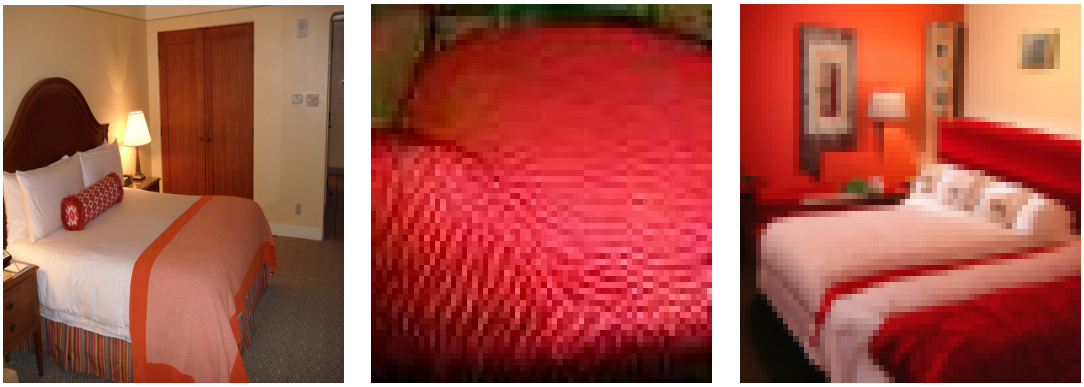

Fig. 4.2.18: A room with a bed that has white and red pillows.

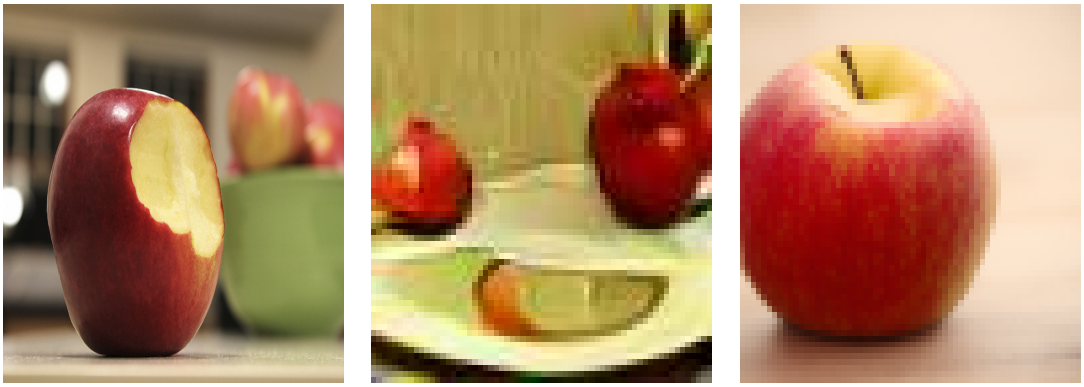

Fig. 4.2.19: A apple with a bite in it on a table.

Fig. 4.2.20: Two park benches that are overlooking a valley.

4.2.3 SOA Scores

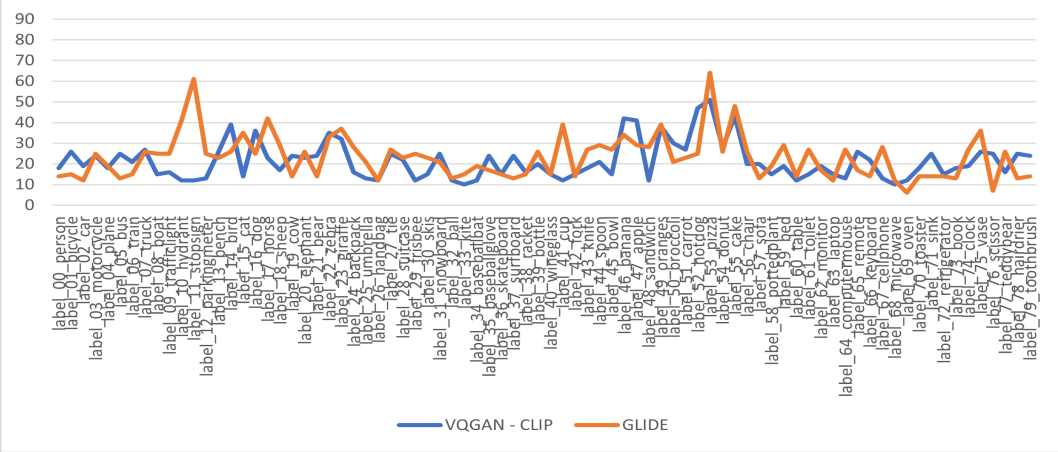

Fig. 4.2.21: SOA Scores for VQGAN - CLIP and GLIDE

We can observe that, for only a few classes, the models are able to recreate the objects mentioned in the captions. Most of the sentences used for the generation contain more than one object, and the objects might sometimes be interacting with each other, which makes it even more challenging for the model. The VQGAN - CLIP is able to generate items like “pizza”, “hotdog”, “cake”, and “banana”, which are common objects. If we had a caption in which the objects interact with other objects the models may not be able to create images this well as observed for other classes. For GLIDE, “pizza” , “cake” , “stop sign” and “horse” are on top. The images in GLIDE also provide more variety than the VQGAN - CLIP model. The

images generated by the VQGAN model appear very similar the images generated by GLIDE appear different for the three cases.

VQGAN - CLIP Generated GLIDE Generated

Fig. 4.2.22: Generated Images: Pizza

Fig. 4.2.23: Generated Images: Cake

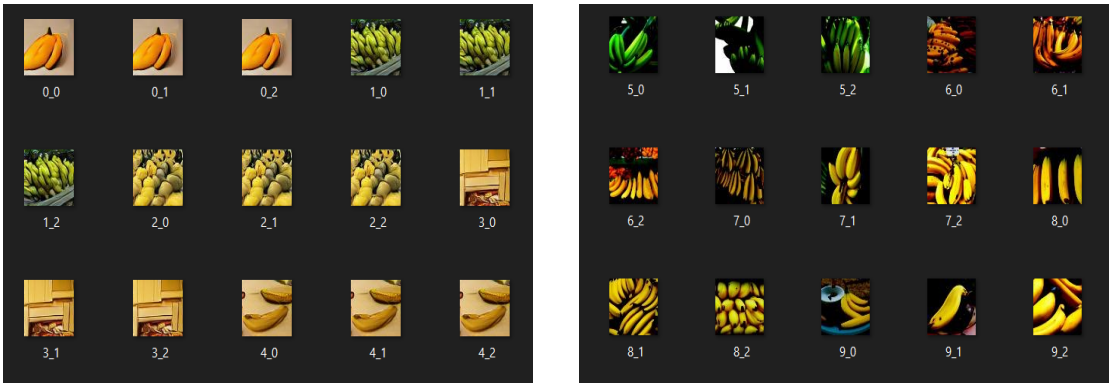

Fig. 4.2.24: Generated Images: Bananas

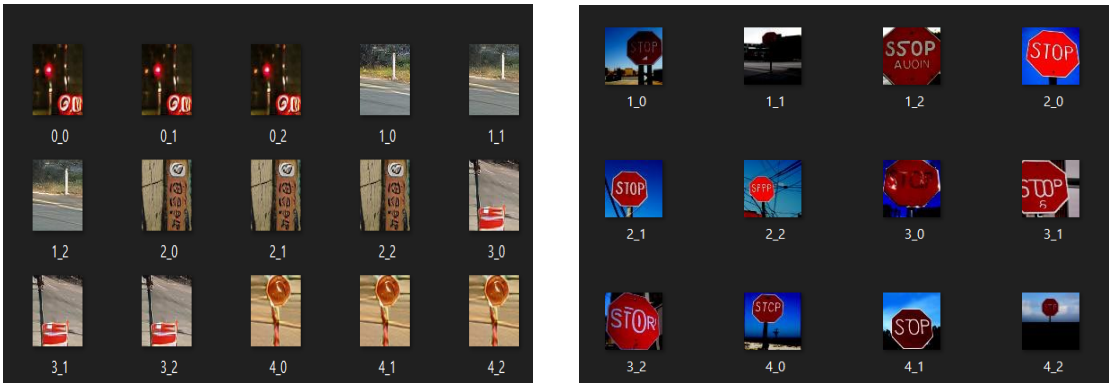

Fig. 4.2.25: Generated Images: Stopsign

Model Name	SOA-C ↑	SOA-I ↑
GLIDE	25.83	32.29
VQGAN - CLIP	23.73	29.67

Table 4.2.2: SOA Scores

Both models perform better on common items but have trouble producing uncommon ones. There is a difference between the SOA-I and SOA-C values. Since the SOA-I is based on the average of images, it is skewed by things that are frequently seen in captions and images, which is why the SOA-I values are higher [18]. GLIDE performs better for both cases proving that there is higher the occurrence of the object in the total images generated. Nevertheless, it is more challenging to construct complicated statements, which are statements with multiple objects in the sentence, using common things than simple phrases with unusual objects.

4.3 Analysis

There are several reasons why a generative model might not be able to generate a certain object:

- Insufficient training data: Generative models require a large amount of training data in order to learn to generate realistic images. If the training data does

not contain examples of the object that the model is being asked to generate, it may not be able to generate that object.

- Unbalanced training data: If the training data is unbalanced (e.g., it contains a disproportionate number of examples of certain types of objects), the model may be biased towards generating those types of objects and may have difficulty generating other types of objects.

- Limited model capacity: Generative models have a limited capacity and may not be able to generate all possible objects if they are too complex or varied. For example, a model trained on a dataset of small, simple images may not be able to generate large, detailed images.

- Overfitting: If the model overfits to the training data (i.e., it performs well on the training data but poorly on new, unseen data), it may not be able to generate objects that are not present in the training data.

For both models, we can deduce that there is lack of training data when the object to be generated is small in size for example “ball” or “baseball bat”. These items are often referred as a small part of a scenery in background. It is tough for the model to understand and generate with respect to the scenery. More emphasis is required on training models with images which have more than two objects and a background. It is also important that correct captions are given while training otherwise the model might learn to associate a cat as some other object. The model performs good if the focus is on one or two objects like “apple” or “pizza” which take up most of the space in the image generated. We can observe that for most of objects with high SOA scores the models are overfitted. The models will only generate the object for which it has most training examples, ignoring other parts of the sentence. For example, if the caption is “A person eating pizza and salad at a table.”. The model will recognize “pizza” immediately and generate an image just containing a pizza (Fig. 4.3.2), which gives us a high SOA but is not semantically accurate image.

Both the models focus on the caption used for the generation of the image and use it to guide the image generation process. It is important that the caption used

contains enough details about the image to be generated. The sentence should have correct spellings of objects for the model to understand as it was trained with correct examples. A very complex sentence with more than two objects makes the models generate images and merge them without focusing on interaction between the objects. The resulting image is distorted which will give a low IS, large FID Score and low SOA Score (Fig. 4.3.1). If the model is trained with enough examples for complex sentences the models may be able to generate better samples.

Fig. 4.3.1: A boy swinging a baseball bat at a ball.

Fig. 4.3.2: Generated Image: A person eating pizza and salad at a table.

CHAPTER 5

Conclusion and Future Work

Although there has been significant progress, more can still be done to improve automated analytics and standardize research. There is still more work to be done before the models can produce images of higher quality that more closely resemble the semantics of the input text. While the models are pretty good at recreating art work and imaginative statements it's still a challenge to be able to recreate multiple objects in a picture from text to image. Even for data sets like CUB or Flowers data set it is relatively easier to recreate as, for most part, they only contain single objects and most of the image created is occupied by them. We can also observe that the objects which have a clear pattern are also easier to detect by the object detector used and can take the texture for object making SOA unreliable for some instances. Although, both models are designed to generate high-quality images from text descriptions, and which one is better depends on the specific task. For our task of generating images from the MS COCO data set the GLIDE model performs better than the VQGAN - CLIP. As a result of our experiments, we were able to determine that the diffusion model GLIDE performed much better for IS and for SOA. While VQGAN - CLIP did perform slightly better for FID, the samples generated by the GLIDE model are semantically accurate and have good IS and SOA scores. The peaks and lows in the results for FID (Fig. 4.2.6) tells us which class objects have more common pixel representation with original data set. This information can be used to determine for which classes the models need more training and attention. The samples generated by GLIDE are very different from the original image giving a high FID score. Given all of this evidence, we can claim that the GLIDE model performs better than the

VQGAN - CLIP model. The images generated by the GLIDE model offers variety by generating unique each time for the same text input. Although there is not a measure to consider how much variety a generative model can offer, we can observe that the images generated by the GLIDE offers more variety when generating a new data set. Both the models are over fitted and biased towards some object classes due to disproportionate number of images in the training data as. For instance, the generated images for pizza the models ignore the other items in the caption and instead focus on generating just pizza pattern in a different way. There is also currently a lack of clarity on the extent to which generative models memorize the training data.

5.1 Future Work

For future work, the researchers can generate images utilizing new approaches such as DALL-E 2 and Imagen for comparison. In order to obtain an even more accurate assessment, the size of the image data set for each class can be increased. The evaluation of the models can benefit from the introduction of new matrices that quantify qualities like variation in generated images. More research can be done to understand how the model memorizes training data. There can be more variation in the data set used. The captions can be taken directly from the people instead of using pre-defined captions. Generative model assessment is critical, and we need this information in order to comprehend the direction in which more work is required.

REFERENCES

[1] Barratt, S. and Sharma, R. (2018). A note on the inception score. *arXiv preprint arXiv:1801.01973*.

[2] Borji, A. (2019). Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 179:41–65.

[3] Brownlee, J. (2019). How to implement the frechet inception distance (fid) for evaluating gans. `https://machinelearningmastery.com/how-to-implement-the-frechet-inception-distance-fid-from-scratch/`. Last accessed 19 December 2022.

[4] Chow, A. (2021). Nfts are shaking up the art world—but they could change so much more. `time.com/5947720/nft-art/`. Last accessed 21 December 2022.

[5] Crowson, K., Biderman, S., Kornis, D., Stander, D., Hallahan, E., Castricato, L., and Raff, E. (2022). Vqgan-clip: Open domain image generation and editing with natural language guidance. In *European Conference on Computer Vision*, pages 88–105. Springer.

[6] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

[7] Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142.

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

[9] Dhariwal, P. and Nichol, A. (2021). Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794.

[10] Esser, P., Rombach, R., and Ommer, B. (2021). Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12873–12883.

[11] Fong, J. (2022). The text-to-image revolution. `https://www.vox.com/23150422/text-to-image-ai-deep-learning`. Last accessed 29 December 2022.

[12] Foster, D. (2019). *Generative deep learning: teaching machines to paint, write, compose, and play*. O'Reilly Media.

[13] Frolov, S., Hinz, T., Raue, F., Hees, J., and Dengel, A. (2021). Adversarial text-to-image synthesis: A review. *Neural Networks*, 144:187–209.

[14] Galatolo, F. A., Cimino, M. G., and Vaglini, G. (2021). Generating images from caption and vice versa via clip-guided generative latent space search. *arXiv preprint arXiv:2102.01645*.

[15] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11):139–144.

[16] Gregor, K., Danihelka, I., Graves, A., Rezende, D., and Wierstra, D. (2015). Draw: A recurrent neural network for image generation. In *International conference on machine learning*, pages 1462–1471. PMLR.

[17] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Klambauer, G., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a nash equilibrium. *arXiv preprint arXiv:1706.08500*, 12(1).

[18] Hinz, T., Heinrich, S., and Wermter, S. (2020). Semantic object accuracy for generative text-to-image synthesis. *IEEE transactions on pattern analysis and machine intelligence*.

[19] Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851.

[20] Karagiannakos, S. and Adaloglou, N. (2022). How diffusion models work: the math from scratch. `Howdiffusionmodelswork:Themathfromscratch,AISummer/`. Last accessed 28 December 2022.

[21] Kingma, D. and Welling, M. (2013). Auto-encoding variational bayes. *ICLR*, pages 1312–6114.

[22] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

[23] Liu, Z., Luo, P., Wang, X., and Tang, X. (2014). Deep learning face attributes in the wild. pages 3730–3738.

[24] Mansimov, E., Parisotto, E., Ba, J., and Salakhutdinov, R. (2015). Generating images from captions with attention.

[25] Miranda, L. J. (2021). The illustrated vqgan. `https://ljvmiranda921.github.io/notebook/2021/08/08/clip-vqgan/`. Last accessed 30 December 2022.

[26] Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. (2021). Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*.

[27] Nichol, A. Q. and Dhariwal, P. (2021). Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR.

[28] Nilsback, M.-E. and Zisserman, A. (2008). Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE.

[29] Oleszak, M. (2022). Autoencoders: From vanilla to variational. `https://towardsdatascience.com/autoencoders-from-vanilla-to-variational-6f5bb5537e4a`. Last accessed 31 December 2022.

[30] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

[31] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.

[32] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021). Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR.

[33] Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

[34] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. (2016). Generative adversarial text to image synthesis. In *International conference on machine learning*, pages 1060–1069. PMLR.

[35] Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR.

[36] Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., et al. (2022). Photorealistic

text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*.

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

[38] Skelton, J. (2022). Generating and editing photorealistic images from text-prompts using openai's glide. `https://blog.paperspace.com/glide-image-generation/#architecture`. Last accessed 31 December 2022.

[39] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR.

[40] Song, J., Meng, C., and Ermon, S. (2020). Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

[41] Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. J. (2011). The caltech-ucsd birds-200-2011 dataset.

[42] Weng, L. (2021). What are diffusion models? `https://lilianweng.github.io/posts/2021-07-11-diffusion-models`. Last accessed 29 December 2022.

[43] Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., and He, X. (2018). Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324.

[44] Yu, Y., Zhang, W., and Deng, Y. (2021). Frechet inception distance (fid) for evaluating gans.

[45] Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., and Metaxas, D. N. (2017). Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915.

VITA AUCTORIS

NAME:	Nazia Siddiqui
PLACE OF BIRTH:	Pratapgarh, India
YEAR OF BIRTH:	1998
EDUCATION:	Andhra Education Society School, New Delhi, India High School, CBSE, 2013-2015
	Jamia Hamdard, New Delhi, India, Bachelor of Technology, Computer Science, 2015-2019
	University of Windsor,Windsor ON, Canada M.Sc in Computer Science, 2021-2022