Text-to-Image Generation

Kirti Motwani[1], Joshuva Jeemon[2], Supriya Mohite[3], Shubham Karande[4], Gautam Gupta[5]
[1,2,3,4,5]*Computer Engineering, Xavier Institute of Engineering,* Mumbai, India

***Abstract*—Machine Learning enables near-perfect algorithmic compositions. The proposed solution, Stacked Generative Adversarial Networks, generates photo-realistic images from text descriptions by decomposing the problem into manageable sub-problems through a sketch-refinement process. The Stage-I GAN sketches low-resolution images of the object's primitive shape and colors. The Stage-II GAN generates high-resolution images with photo-realistic details by rectifying defects in Stage-I results and adding compelling details with the refinement process. A Conditioning Augmentation technique improves diversity and stabilizes training. The proposed method achieves significant improvements in generating photo-realistic images conditioned on text descriptions.**

***Index Terms*—Generator, Discriminator, Generative adversarial networks, Conditioning augmentation.**

I. INTRODUCTION

Infographics can make it easier to share information in a visually appealing and engaging manner on websites, blogs, and social media. They can be quickly read and are attention-grabbing, making them a useful tool for communicating complex information in a user-friendly way. However, creating infographics manually can be time-consuming. To address this issue, there is a growing interest in using artificial intelligence and machine learning algorithms to automatically generate images based on natural language descriptions. While the state-of-the-art algorithms have made significant progress in this area, the task of synthesizing realistic images remains challenging.

When people hear or read a story, they create mental images in their minds to better understand the content. The ability to visualize is essential for cognitive functions like memory, reasoning, and thinking. Creating technology that can connect written descriptions with visual images is a significant step towards enhancing intellectual ability. Text can sometimes be difficult to understand or misinterpreted, but images can make it easier to comprehend. Visual aids can convey information more effectively and engage people's attention. Presentation, learning, and many other activities benefit from visual communication. We used the deep learning technique Generative Adversarial Network (GAN) with tensorflow, numpy, and tensorlayer to generate images from text. Natural Language Toolkit (NLTK) was used to divide larger texts into smaller words for analysis. This project can save countless hours for those without artistic skills, making it beneficial for fields such as design, infographics, modeling, and creative compositions. [2] A GAN-based model can produce high-quality images that capture fine-grained information at the word level and are visually pleasing to look at. The task of creating high-resolution images from textual descriptions is difficult but crucial for practical applications such as art generation and computer-aided design. With the emergence of deep generative models, there has been significant progress in this area. In addition to using a single GAN to generate images, there has also been work done using a series of GANs for image generation.

II. REVIEW OF LITERATURE

Generative Adversarial Networks, are an approach to generative modeling using deep learning methods, such as convolutional neural networks. Generative modeling is an unsupervised learning task in machine learning that involves automatically discovering and learning the regularities or patterns in input data in such a way that the model can be used to generate or output new examples that plausibly could have been drawn from the original dataset. GAN are a clever way of training a generative model by framing the problem as a supervised learning problem with two sub-models: the generator model that we train to generate new examples, and the discriminator model that tries to classify examples as either real (from the domain) or fake. The two models are trained together in a zerosum game, adversarial, until the discriminator model is fooled about half the time, meaning the generator model is generating plausible examples. GAN are an exciting and rapidly changing field, delivering on the promise of generative models in their ability to generate realistic examples across a range of problem domains, most notably in image-to-image translation tasks such as translating photos of summer to winter or day to night, and in generating photorealistic photos of objects, scenes, and people that even humans cannot tell are fake.

Types of GAN:

- Vanilla GAN

The Vanilla GAN is the simplest type of GAN made up of the generator and discriminator, where the classification and generation of images is done by the generator and discriminator internally with the use of multi-layer perceptronss. The generator captures the data distribution meanwhile, the discriminator tries to find the probability of the input belonging to a certain class, finally the feedback is sent to both the generator and discriminator after calculating the loss function, and hence the effort to minimize the loss comes into picture.

- Attentional GANs

Attentional GAN is a variant of the Generative Adversarial Network (GAN) architecture that incorporates attention mechanisms to improve the quality of generated images. Attention mechanisms allow the model to focus on specific regions of an image during the generation process, which can improve the realism and fine-detail of the generated image. Attentional GANs have been shown to produce high-quality images in various image generation tasks such as image synthesis, style transfer, and super-resolution. Some examples of Attentional GANs include Attention GAN, Self- Attention GAN, and Recurrent Attention GAN.

- Stack GANs

StackGANs (Stacked Generative Adversarial Networks) are a class of generative models that can generate high-resolution images by progressively refining the details of the image at multiple scales. StackGANs were first introduced in a research paper titled” StackGAN: Text to Photorealistic Image Synthesis with Stacked Generative Adversarial Networks” by Han Zhang et al. in 2017. The basic idea behind StackGANs is to use a two-stage architecture that involves generating low-resolution images first and then progressively refining them to generate higher-resolution images. The first stage of the StackGAN generates a low-resolution image from a textual description using a conditional GAN (Generative Adversarial Network). The second stage of the StackGAN takes the low-resolution image as input and generates a higher-resolution image with finer details.

- DF-GANs

DF-GAN (Deep Fusion Generative Adversarial Network) is a generative model for image synthesis that was introduced in a research paper titled” Deep Fusion Generative Adversarial Networks for Text-to-Image Synthesis” by Zhengli Zhao et al. in 2018[5]. The basic idea behind DF-GAN is to use a deep fusion architecture that combines the strengths of both GANs (Generative Adversarial Networks) and CNNs (Convolutional Neural Networks) to generate high-quality images from textual descriptions. DF-GAN consists of two main components: a generator network and a discriminator network[4]. The generator network takes a textual description as input and generates an image that matches the description. The discriminator network, on the other hand, takes an image and a textual description as input and predicts whether the image matches the description or not. The two networks are trained in an adversarial manner, where the generator network tries to generate images that fool the discriminator network, and the discriminator network tries to distinguish between real and generated images.

- VQ-GAN

VQGAN (Vector Quantized Generative Adversarial Network) is a generative model for image synthesis that combines the strengths of GANs (Generative Adversarial Networks) and VQ-VAE (Vector Quantized Variational Autoencoder). VQGAN was introduced in a research paper 15 titled ”Making Efficient Use of Deep Generative Models for High-Resolution Image Synthesis” by Patrick Esser et al. in 2021. [7] The basic idea behind VQGAN is to use a two-stage architecture that involves encoding the input image into a discrete latent space using VQ-VAE and then decoding the latent space into a high-resolution image using a GAN. The discrete latent space allows for efficient and effective representation of complex image features, while the GAN enables high-quality and diverse image synthesis.

III. PROPOSED SYSTEM

Stack GAN Model:
StackGAN is a generative adversarial network (GAN) architecture that was introduced in 2017 for synthesizing high-resolution, realistic images. The architecture is designed to generate images from text descriptions, allowing users to generate realistic images of objects or scenes described in natural language.
StackGAN operates in a two-stage process. In the first stage, it generates a low-resolution image based on the text description. In the second stage, it takes the low-resolution image and the text description as inputs, and generates a high-resolution image.
The key innovation of StackGAN is its use of a conditioning augmentation technique that injects the text description into the GAN at multiple stages of the network, allowing the generator to receive more detailed information about the desired output image. Additionally, the network uses a multi-scale approach that generates images at different resolutions, which helps to capture both global and local image features. [4] StackGAN has been shown to generate high-quality images that are difficult to distinguish from real images. It has been used for various applications, including generating images of birds, flowers, and bedrooms. The architecture has also been extended to a three-stage StackGAN++, which further improves the quality and diversity of generated images.
This formulation's brilliance lies in the antagonistic relationship between the Generator and the Discriminator. When a fake sample (which is produced by the Generator) is presented to a discriminator, it wants to identify it as such and call it out as such. But, the Generator wants to produce samples in such a way that the discriminator calls it out as such by mistake. In a way, the Generator is attempting to deceive the Discriminator.

Step 1:
Text Encoder: The text encoder network typically uses a pre-trained language model such as BERT or GPT to extract meaningful features from the input text. These features are then fed through a series of fully connected layers to produce the final text embedding. The text encoder plays a critical role in StackGAN by capturing the semantic meaning of the input text and enabling the generator network to generate high-quality images that are consistent with the input description. [1] The quality of the text encoder greatly impacts the overall performance of the StackGAN model, as a good text encoder should be able to capture the key attributes and characteristics of the input text that are relevant for generating the corresponding image.

Step 2:
Embedding Vector - The embedding vector is a fixed-length vector representation of the input text, which captures the semantic meaning of the text. The text-embedding network typically uses a pre-trained language model such as BERT or GPT to extract features from the input text and map it to the embedding vector. In StackGAN, the embedding vector is first fed to a conditioning augmentation module, which applies noise to the embedding vector to introduce stochasticity and increase diversity in the generated images. The resulting conditioned embedding vector is then fed to the image synthesis network, which generates the corresponding image.

Step 3:
Conditioning Augmentation - The technique involves generating realistic data samples by conditioning the model on a given set of information or attributes. One popular method to achieve this is to use embedding vectors, which encode the attributes of the sample in a lower-dimensional space. [5] By injecting these embedding vectors into the model, it can generate samples that conform to the given attributes. However, embedding vectors alone may not be sufficient to produce diverse and realistic samples. To address this issue, random noise can be added to the embedding vectors, creating a continuous latent space that allows for interpolation between attributes and noise vectors. This results in a larger space of possible outputs, allowing for greater diversity and richness in the generated samples.

Step 4:
Stage 1 GAN - GAN stands for Generative Adversarial Networks, which is a type of machine

learning model used for generating new data. Stage 1 GAN is the first phase in a two-stage GAN training approach that involves training a generator and discriminator network simultaneously.

In Stage 1 GAN, the generator network is trained to generate images that resemble the training data, while the discriminator network is trained to distinguish between real and fake images. The generator takes a random noise vector as input and generates an image, while the discriminator takes an image as input and predicts whether it is real or fake. During training, the generator tries to generate images that can fool the discriminator into thinking they are real, while the discriminator tries to correctly identify whether an image is real or fake. This adversarial training process continues until the generator can produce images that are indistinguishable from the real training data.

Step 5:

Stage 2 GAN - Stage 2 GAN is the second phase of a two-stage GAN training approach, which is used to generate high-quality images with fine details and textures. In Stage 2 GAN, a pre-trained generator from Stage 1 is fine-tuned using a new dataset that includes higher resolution images. The training process involves adding new layers to the generator network and increasing its capacity to generate more complex images. The discriminator network from Stage 1 is also retrained to evaluate the new high-resolution images.

During training, the generator network is optimized to generate images that not only look real but also have fine details and textures. The discriminator network is optimized to distinguish between real and fake images at the higher resolution. Stage 2 GAN typically requires more computational resources and training time compared to Stage 1 GAN, but it results in higher quality generated images that are suitable for applications such as high-resolution image synthesis, super-resolution, and texture synthesis.

One approach to generative modeling employing deep learning techniques, such as convolutional neural networks, is known as generative adversarial networks, or GANs.

Generative modeling is a machine learning task that involves automatically identifying and learning the regularities or patterns in input data so that the model can be used to produce new examples that could have been reasonably derived from the original dataset. By framing the challenge as a supervised learning problem with two sub-models—the generator model, which we train to create new instances, and the discriminator model, which tries to categorize examples as either real (from the domain) or fake—GANs are a creative method to train a generative model (generated). The discriminator model is tricked roughly half the time during training of the two models, indicating that the generator model is producing believable examples.

IV. IMPLEMENTATION METHODOLOGY

Generative Adversarial Networks (GANs) employ deep learning techniques, specifically convolutional neural networks, for generative modeling. This unsupervised machine learning task involves automatically identifying patterns in input data to output new, plausible examples. GANs use a generator model that takes a fixed-length random vector as input and produces samples in the domain by generating data from a compressed representation of the original data distribution. Additionally, a discriminator model predicts whether an input is real or fake (generated). Once trained, the discriminator is discarded, and the focus remains on the generator.

GANs follow a Learn-Generate-Improve process to create anything fed to them. [2] To comprehend GANs, knowledge of Convolutional Neural Networks (CNNs) is necessary. CNNs are trained to classify images by analyzing pixels and identifying them based on hidden layers. Conversely, GANs consist of two parts: the Generator and Discriminator. The Discriminator acts like a CNN and differentiates between real and fake data, outputting 1 or 0 depending on the data's authenticity. The Discriminator learns to recognize features of real data to classify data correctly.

The approach shown in **Figure 1** for the entire process is as follows:

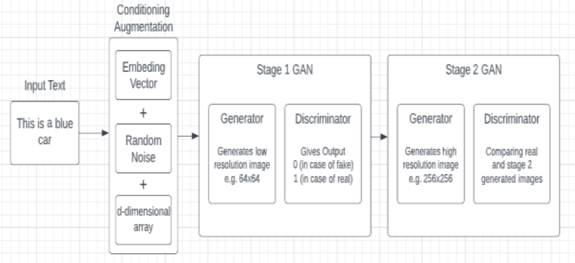
Input Text
This is a blue car
Conditioning Augmentation
Embeding Vector
+
Random Noise
+
d-dimensional array
Stage 1 GAN
Generator
Generates low resolution image e.g. 64x64
Discriminator
Gives Output 0 (in case of fake) 1 (in case of real)
Stage 2 GAN
Generator
Generates high resolution image e.g. 256x256
Discriminator
Comparing real and stage 2 generated images

Figure 1: Block diagram of the process

Step 1: User Input Text - As per the desire of the user, they input textual information as required.

Step 2: Text Embedding via Vectors - Embeddings are learned representations of Text as a vector of numbers. Word2Vec is a statistical method for efficiently learning a standalone word embedding from a text corpus. It converts the textual information into numbers such as the notion of relatedness across words or products such as semantic relatedness, synonym detection, concept categorization, selectional preferences, and analogy.

Step 3: Conditioning Augmentation - A conditioning augmentation (CA) network samples random latent variables from a distribution, which is represented as . We will learn more about this distribution in later sections. There are many advantages to adding a CA block, as follows:

- It adds randomness to the network.
- It makes the generator network robust by capturing various objects with various poses and appearances.
- It produces more image-text pairs. With a higher number of image-text pairs, we can train a robust network that can handle perturbations.

Retrieving the numeric values from the embedding vector and storing them in a d-dimensional array. This d-dimensional array is now cumulated with some ”Random Noise”. These 3 entities together form our Conditional Augmentation Cycle.

Step 4: The Stage 1 GAN Sketches the primitive shape and basic colors of the object.

Step 5: The Stage 2 GAN Corrects defects in the low-resolution image from Stage I and completes details of the object by reading the text description again, producing a high-resolution photo-realistic image.

V.CONCLUSION

The current project can be implemented into a more advanced version of itself with more complex computation and larger dataset. The future scope of extending the project can be done in 2 main ways:-

Improved Quality: The quality of generated images can be improved by developing more advanced generative models such as GANs, which can generate high-quality images with fine details and textures.

Multimodal generation: Generating images from textual input is just one aspect of multimodal generation. It is possible to extend this to other modalities such as audio, video, and speech, which could have applications in virtual reality, gaming, and content creation.

The intersection of artificial intelligence and the fine arts is currently at an exciting stage, as both fields are influencing and inspiring each other. The use of creative images has become increasingly popular, with applications ranging from advertising to digital art. Through extensive research, we have explored the topic of generating images from text and compared various models, such as the Minimal Dall-E and Stable-Diffusion Models. However, many of these models either require payment or lack accuracy and efficiency. As a result, our future goal is to develop an alternative model that can generate high-quality images from user-provided data in a timely and cost-effective manner.

REFERENCES

[1] Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B. and Bharath, A.A., 2018. Generative adversarial networks: An overview. IEEE signal processing magazine, 35(1), pp.53-65

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

[3] Liu, M.Y., Huang, X., Yu, J., Wang, T.C. and Mallya, A., 2021. Generative adversarial networks for image and video synthesis: Algorithms and applications. Proceedings of the IEEE, 109(5), pp.839-862.

[4] 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), pp.139-144.

[5] Liang, S., Li, Y. and Srikant, R., 2017. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*.

[6] Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X. and Metaxas, D.N., 2018. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE transactions on pattern analysis and machine intelligence*, *41*(8), pp.1947-1962.

[7] Yu, J., Li, X., Koh, J.Y., Zhang, H., Pang, R., Qin, J., Ku, A., Xu, Y., Baldridge, J. and Wu, Y., 2021. Vector-quantized image modeling with improved VQGAN. *arXiv preprint arXiv:2110.04627*.