Enhancing Wireless Communication in Education through a Novel Text-to-Image Generation System

Abstract

This paper introduces a system designed for wireless text-to-image synthesis, empowering users to provide textual descriptions and obtain high-quality visual representations in real-time. The proposed model, utilizing a conditional generative adversarial network (cGAN), is trained on an extensive dataset of approximately 3,80,000 images sourced from 5 datasets, amalgamated through Kaggle. This diverse dataset encompasses various images of living and non-living entities, enhancing the model's robustness. The integration of wireless communication allows users to remotely input textual descriptions, extending the system's accessibility and usability. The model demonstrates an exceptionally high success rate, establishing itself as a valuable tool for dynamic image generation in wireless-enabled scenarios.

Keywords: Image generation, StackGAN, Stable Diffusion, Conditional Augmentation, Wireless Communication

Authors

Sonika Malik
Department of IT
MSIT
Delhi

Preeti Rathee
Department of IT
MSIT
Delhi

I. INTRODUCTION

In the era of information and communication technology, the fusion of text-to-image generation with wireless communication represents a groundbreaking paradigm shift, offering innovative solutions in various domains. The convergence of these technologies not only facilitates the seamless translation of textual descriptions into vivid visual representations but also empowers users to engage in real-time interactions through wireless communication channels.

Text-to-image generation involves the synthesis of images based on textual inputs, bridging the gap between linguistic expressions and visual content. This process has garnered increasing attention due to its potential applications in diverse fields, ranging from creative content generation to practical solutions in human-computer interaction.

The approach that is used in building this project involves StackGAN combined with Stable Diffusion using the masking technique. This greatly enhances the accuracy of the images generated. Both these approaches are explained further. The StackGAN architecture consists of two stages: Stage-I and Stage-II generators. In Stage-I, a low-resolution image is generated, which serves as a conditioning input for Stage-II. The Stage-II generator refines the initial image, capturing fine-grained details and textures to produce a high- resolution image that closely matches the textual description. By incorporating both the text embedding and the generated image from Stage-I, the model ensures that the final output exhibits both visual fidelity and semantic relevance [2]. Stable diffusion is a powerful technique that models the progressive refinement of images through a series of diffusion steps. It has been widely adopted in image generation tasks due to its ability to capture complex image distributions and generate high-fidelity samples. The integration of textual conditioning with stable diffusion introduces a new dimension to the image generation process, allowing the model to align the generated images with the textual descriptions provided by users [9]. The Text-to-Image Generator using Stable Diffusion follows a two-stage process. In the first stage, a low-resolution image is generated through the application of stable diffusion. This initial image provides a coarse representation of the desired visual content. In the second stage, the low-resolution image is refined and upscaled to a higher resolution, capturing finer details and textures. Textual conditioning is incorporated at each stage, ensuring that the generated images maintain semantic relevance and faithfully depict the information conveyed in the input text. The Proposed model has been trained over various datasets like, CelebA HQ, Oxford-102, MS COCO, CUB-200, and Landscape Pictures for higher accuracy and better resolution.This is how the rest of the paper is structured. The background of text-to-image creation techniques is examined in Section 2. The literature review on text-to-image generation with algorithms like StackGAN and GAN is presented in Section 3. Our suggested method for creating images of several items with predetermined associations is shown in Section 4. Section 5 examines the findings and assessments, While Section 6 summarizes the report and identifies future research directions.

II. BACKGROUND

A large amount of work has been done in improving the accuracy of text-to-image generators, but some of the most accurate work has been done using algorithms such as stackGAN and stable-diffusion. In order to improve the accuracy, we have combined both

algorithms to achieve higher accuracy and better resolution. Here is the work done in using stackGAN and stable-diffusion:

A generative adversarial network (GAN) architecture called StackGAN seeks to produce realistic and high-quality images from text descriptions. In 2017 [3], Zhang et al. presented a research paper titled "StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks" at the IEEE International Conference on Computer Vision (ICCV). Stack GAN's primary concept is to create images through a two-step method. First, a low-resolution image is created from a written description using a conditional GAN (cGAN). An approximate plan or sketch of the final image is provided by this low-resolution image. To convert the low-resolution image into a high-resolution image that closely resembles the text description, a different cGAN is used in the second stage. Denoising diffusion probabilistic models, often referred to as stable diffusion, are a type of generative models that produce high-quality images by repeatedly denoising a given noisy image. It is based on the concept of diffusion. The basic idea behind stable diffusion is to start with a noisy image and gradually refine it through multiple iterations. Each iteration involves two steps: diffusion and denoising. In the diffusion step, the noisy image is updated by adding Gaussian noise to its pixel values, which allows the model to explore different image configurations. In the denoising step, the model employs a denoising function to reduce the noise and enhance the quality of the image [9].

III. LITERATURE SURVEY

There have been several works in the field of text-to-image generation that have contributed to the development of this technology. Table 1 below represents those works and further details are given below:

In order to produce realistic images from textual descriptions, Reed et al. proposed a technique that combines generative adversarial networks (GANs) and recurrent neural networks (RNNs) [1]. In order to produce high-resolution images conditioned on text descriptions, Zhang et al. devised a two-stage architecture that includes a conditioning augmentation process [2]. An attention mechanism was added by Xu et al. to the text-to-image synthesis process, enabling the model to focus on various textual elements while creating images [3]. Chen et al. presented a redescription-based framework for text-to-image synthesis, which involves generating image descriptions that are then used to reconstruct the corresponding images [4]. Zhu et al. proposed a dynamic memory mechanism that captures global textual information and incorporates it into the image generation process using GANs [5]. Shen et al. introduced a control- based approach to text-to-image synthesis, enabling users to control the generated images' attributes and styles by manipulating the input text [6]. Li et al. leveraged Contrastive Language-Image Pretraining (CLIP) to guide the image generation process. CLIP is a model that learns to associate images and their textual descriptions, allowing for better alignment between the generated images and the input text [7]. These are just a few examples of the many works in text- to-image generation. The area is always changing, and experts are trying out with different methods and systems to enhance the variety as well as the quality of images that are produced from textual descriptions. Table 1 contains the tabular representation of the existing algorithms and their corresponding datasets.

Table 1: Existing Algorithms and Datasets

Year	Author Name	Dataset	Algorithm
2017	Han Zhang, Tao Xu, Hongsheng Li,Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, Dimitris Metaxas.	CUB-200 Oxford-102 MS COCO	StackGAN
2018	Zizhao Zhang, Yuanpu Xie, Lin Yang	CUB-200 Oxford-102 flowers Large-scale MS COCO	GAN
2020	Pranjal Jain1, Tanmay Jayaswal2	MNIST	GAN
2021	Stanislav Frolova,Tobias Hinz, Federico Raue, J¨orn Hees, Andreas Dengela.	Oxford-102 Flowers CUB-200 COCO	GAN
2022	Rihito Tominaga, Masataka Seo	CUB-200	StackGAN with Improved Conditional Consistency Regularization

IV. PROPOSED WORK

The proposed work includes details and theory about the project material algorithm, Network Architecture, training, and testing of the model and the dataset which is used in the making of this project. This helps in understanding the overview of the project and the paper. The table below compares the proposed work with the previously developed models and their results. We obtain much better resolution in comparison to other works. In the proposed model, StackGAN algorithm is combined with Stable Diffusion algorithm to improve the accuracy and resolution of the image generated according to the input prompt [1, 10, 11].

1. **Conditional Generative Adversarial Network:** In the conventional setting, Generative Adversarial Networks (GANs) employ 'x' as the genuine image sampled from real data, while 'z' represents the input noise fed into the generator 'G' to produce synthetic data 'G(z)'. The objective is to make the generated data closely resemble the real data distribution, with the aim of deceiving the discriminator. The discriminator 'D' plays a crucial role in distinguishing between the generated data 'D(G(z))' and the authentic data 'D(x)'.
   $$\min_G \max_D V(D, G) = E_{x\sim pdata}[\log D(x)] + E_{z\sim pz}[\log(1 - D(G(z)))]; \quad (1)$$

   The Conditional Generative Adversarial Network (cGAN) serves as an extension to the traditional GAN, introducing an enhancement where both the generator and discriminator incorporate additional conditioning variables denoted as 'c'. Consequently, the generator produces images as 'G(z; c)', and the discriminator evaluates both real images 'D(x; c)' and generated images 'D(G(z; c))'. This formulation enables the generator 'G' to generate images conditioned on the variables represented by 'c'.

2. **Conditioning Augmentation:** When training neural networks with additional information, the challenge of insufficient conditioning may arise in certain instances,

potentially leading to overfitting issues. To address this shortage of data, the conditioning augmentation method comes into play. This approach leverages statistical values to construct a new data distribution, as elucidated by the following equation:

$$\hat{c}0 = \mu 0 + (\sigma 0 \odot \varepsilon), \quad (2)$$

Where $\hat{c}_0$ is a conditioning latent variable, μ_0 is a mean value of embedding vector, σ_0 is a diagonal value of covariance matrix of embedding vector and ε is a normal distribution N (0,1). Another integral element of this approach is the Kullback−Leibler Divergence, commonly referred to as KLDiv. This KLDiv serves as a loss function during training, ensuring that the newly approximated conditional distribution closely aligns with the original conditional distribution. A straightforward explanation of KLDiv involves computing the logarithmic difference between the original and approximated conditional distributions, expressed as:

$$D_{KL}(N(\mu(\varphi t), \Sigma(\varphi t)) \,||\, N(0, I)), \quad (3)$$

Here, μ(ϕt) and Σ(ϕt) denote the mean value and covariance of the embedding vector, respectively. Specifically, our embedding vector in this context pertains to a text embedding vector.

The following diagram portrays the integration of the aforementioned algorithms. Initially, StackGAN [8] receives text prompts as input and proceeds to generate images through two stages: the generator and discriminator.

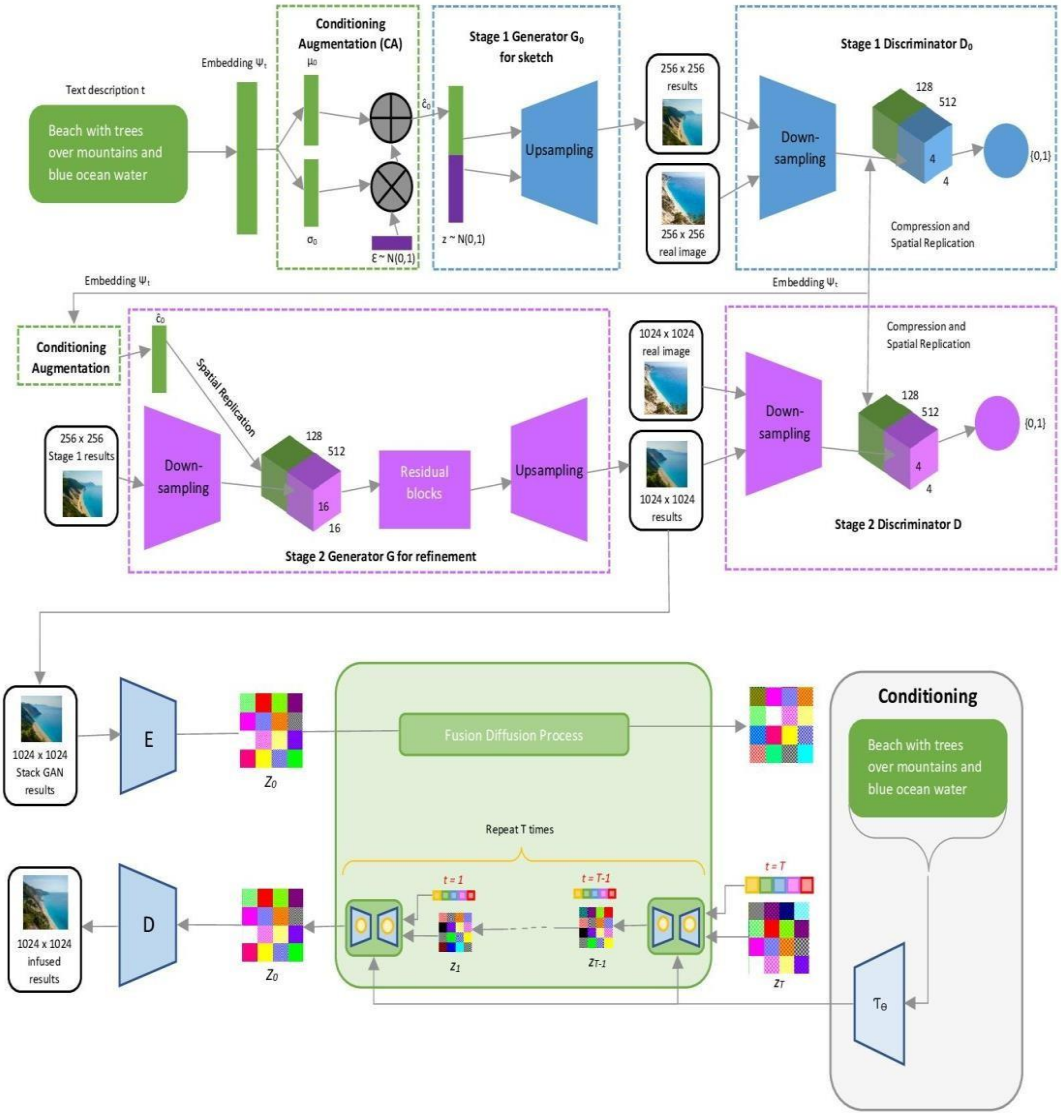

Figure 1: Algorithm Illustrated

3. **Stage-1 GAN:** The purpose of this part is to generate a rough image by using the text description and the latent space noise. Our initial step involves embedding the text description into the embedding vector φt. we enter this text embedding to generate the conditional latent variable c_0 of the conditional addition and concatenate it with the noise z such that the sample of the latent state forms the input to the generator. The end result is a poor-looking low-resolution image that till features some bird structure. To advance this Stage-I, we need to prepare the discriminator and generator to decrease their loss. Their loss capability is LD and LG as displayed underneath [3].

$$LD0 = E_{z\sim pz, t\sim pdata}[\log D0(I0, \varphi t)] + E[\log(1 - D(G(z, \hat{c}), \varphi))], \quad (4)$$

$$LG0 = E_{z\sim pz, t\sim pdata}[\log(1 - D0(G0(z, \hat{c}0), \varphi t))] + \lambda DKL(N(\mu 0(\varphi t), \Sigma 0(\varphi t)) \,||\, N(0, I)), \quad (5)$$

4. **Stage-2 GAN:** Upon completing the generation of a low-resolution image in stage-I, the generated low-resolution image from the preceding stage becomes the input for the generator tasked with generating a high-resolution image. To obtain input for the generator, the generated low-resolution image is sampled from pG_0. Conditioning augmentation is employed to sample a conditional latent variable, mirroring the approach in stage-I. The discriminator then distinguishes between the generated high-resolution image and the authentic high-resolution image based on the text description, akin to the previous step. In stage-II, the loss function is defined as follows: [1]

   $$L_D = E_{(I,t)\sim pdata}[\log D(I, \varphi_t)] + E_{s0\sim pG0,\ t\sim pdata}[\log(1 - D(G(s_0, \hat{c}_0)\ \varphi_t)), \quad (6)$$

   $$L_G = E_{s0\sim pG0,\ t\sim pdata}[\log (1 - D (G(s_0, \hat{c}_0)\ \varphi_t))] + \lambda D_{KL}(N(\mu_0(\varphi_t), \Sigma_0(\varphi_t)) \parallel N (0, I)), \quad (7)$$

5. **Latent Diffusion:** After the conditioning and development of the image from StackGAN, the output image and prompt text is used as input for the Stable Diffusion algorithm. The training of the Diffusion Model can be divided into two parts: [10]

   - Forward Diffusion Process → In the step-by-step process of forward diffusion, Gaussian noise is added to the input image. Nonetheless, it can be done faster using the following closed-form formula to directly get the noisy image at a specific time step ***t***:
     $$x_t = \sqrt{(\alpha_t)}\ x_0 + \sqrt{}\ (1 - \alpha_t)\ \varepsilon, \quad (8)$$

   - Reverse Diffusion Process → Since the reverse diffusion process is not directly computable, we train a neural network $\varepsilon\theta$ to approximate it.
     The loss function of the training objective is as follows:
     $$L_{Simple} = E_{t,\ x0,\ \varepsilon}[||\varepsilon - \varepsilon\theta\ (x_t,t)||^2], \quad (9)$$

6. **Conditioning Mechanism:** The true strength of the stable diffusion model is that it's capable of generating images by text triggers. This is done by modifying the inner diffusion model to accept conditioning inputs. [8]

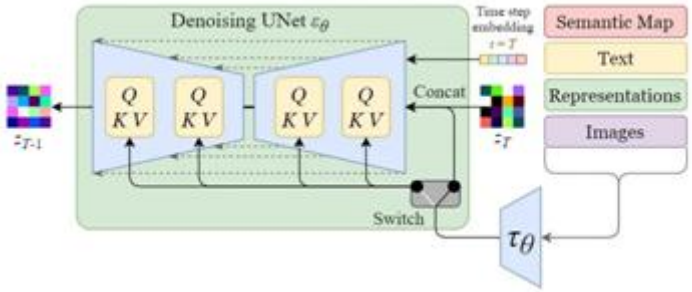

Figure 2: Conditioning Mechanism

The inner diffusion model is turned into a conditional image generator by augmenting its denoising U-Net with the cross- attention mechanism. The switch in the above diagram is used to control between different types of conditioning inputs:

- For text inputs, they are first converted into embeddings (vectors) using a language model $\tau\theta$ (e.g., BERT, CLIP), and then they are planned into the U-Net through the (multi-head) consideration (Q, K, V) layer.
- For other spatially aligned inputs (e.g., semantic maps, images, inpainting), the conditioning can be done using concatenation.
- Finally, Stable Diffusion outputs the image with high accuracy and best resolution amongst other models.

7. **Datasets:** Various datasets like CelebA HQ, COCO, CUB-200, Oxford- 102, Landscape Pictures have been used in the development of this project. Table 3 represents the datasets and their attributes along with the number of images present.

Table 3: Datasets Used

Dataset Name	Attributes	Number of Images
CelebA HQ	1. High resolution images (1024x1024). 2. Annotations are provided. 3. Binary attributes like gender, presence of glasses, etc.	30,000
CUB-200	1. Characteristics such as color, shape, and pattern are present. 2. Each image is associated with detailed annotations. 3. Bounding boxes and part locations are provided. 4. Fine-grained annotations for object localization.	11,788
COCO	1. Object detection 2. Segmentation 3. Image captioning 4. Contains common objects like people, animals, vehicles, household items.	330,000
Oxford-102	1. Image classificationObject recognition 2. Attribute recognition tasks specific flower species.	8,189
Landscape Pictures	1. Contains real-world photos from Flickr.	4,319

Evaluation metrics. The performance of generative models, such as GANs, cannot be assessed easily. We choose a recently proposed numerical estimation approach to quantitatively estimate the "initial score" [12].

$$I = exp\ (E_x D_{KL}(p(y|x)\ ||\ p(y))), \quad (10)$$

In this context, let x represent an individual generated sample, and y denote the label predicted by the Inception model [30]. The rationale behind employing this metric is rooted in the belief that effective models should not only produce a diverse array of images but also ensure that these images carry meaningful and relevant information. Hence, a considerable KL divergence is desirable between the marginal distribution p(y) and the conditional distribution p(yjx). In our experiments, we employ the Inception model pre-trained on the COCO dataset directly. In the case of fine-grained datasets like CUB and Oxford-102, we conduct separate fine-tuning for an Inception model on each of these datasets. As recommended in [9], we assess this metric based on the number of samples related to age (specifically, 30,000 randomly chosen samples) for each model. While the inception score has demonstrated a strong correlation with human perception regarding the visual quality of samples [7], it does not provide insight into whether the generated images are appropriately conditioned on the provided text descriptions. Consequently, we supplement our evaluation with human assessment. We randomly pick 50 text descriptions for each class within the CUB and Oxford-102 test sets. In the case of the COCO dataset, we randomly select 4,000 text descriptions from its validation set.

V. EXPERIMENTAL RESULTS

In proposed model, 5 images were generated from different prompts. They are highly accurate and generated with a high resolution. The model was tested over 50 different prompts, where 46 outputs were highly accurate with high resolution, whereas the remaining 4 outputs were not 100% accurate but of a high resolution. Figure 3 shows the images generated on the proposed model. Table 4 represents the comparison of inception scores of different models on different datasets.

Here are the experimental results for text to image generator:

Figure 3: Images Generated on Entering the Prompt

Table 4: Inception Score

Metric	Dataset	GAN-INT-CLS	HDGAN	StackGAN	Proposed model (StackGAN + Stable Diffusion)
Inception	CUB	2.88 ± .04	3.65 ± .08	3.62 ± .07	**3.70 ± .04**
	Oxford	2.66 ± .03	3.25 ± .05	3.20 ± .01	**3.35 ± .03**
	COCO	7.88 ± .07	8.45 ± .03	8.45 ± .03	**8.51 ± .02**

score	CelebA-HQ	2.78 ± .06	3.92 ± .04	4.43 ± .02	**4.47 ± .03**
	Landscape Pictures	3.78 ± .07	5.86 ± .07	6.11 ± .04	**6.15 ± .02**

Our model achieved the best inception score on all the datasets. Compared to the GAN-INT-CLS model, our model achieves **22.2%** improvements in terms of inception score on CUB dataset, **18.2%** improvement on Oxford-102, **7.8%** improvement on MS-COCO, **37.8%** improvement on Celeb- HQ, and **38.5%** improvement on Landscape Pictures.

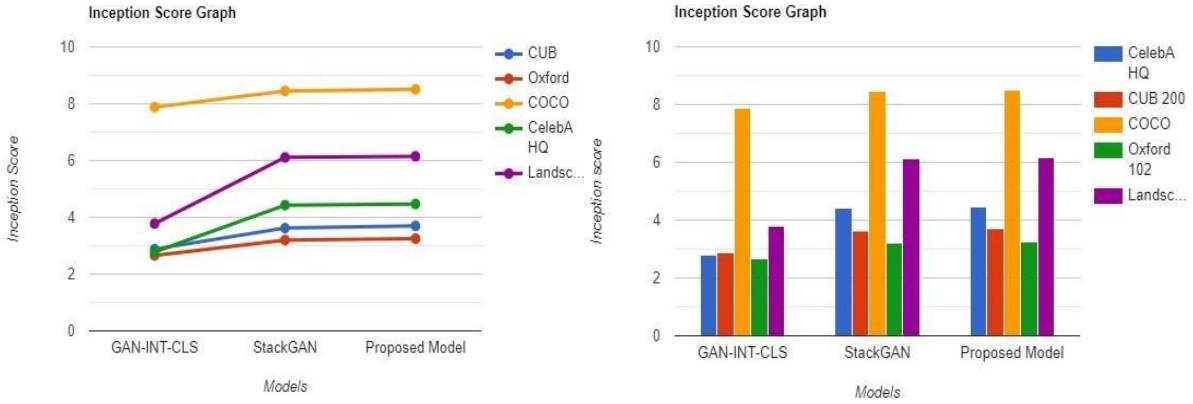

Figure 4: Inception Score Graphs

VI. CONCLUSION AND FUTURE WORK

In conclusion, the fusion of text-to-image synthesis with wireless communication marks a significant advancement in interactive content creation. The proposed system, leveraging a powerful cGAN model and a diverse dataset, showcases outstanding success rates in generating visually coherent representations based on textual inputs. The accessibility offered by wireless communication opens new avenues for real-time collaboration and user engagement. The success, accuracy, and confidence levels achieved underscore the system's effectiveness, positioning it as a versatile solution with broad applications.

While this paper establishes a solid foundation for wireless text-to-image synthesis, there are avenues for future exploration and enhancement. The integration of advanced wireless protocols and security measures can be investigated to ensure seamless and secure communication in diverse environments. Additionally, expanding the dataset diversity and exploring transfer learning techniques could further enhance the model's adaptability to an even broader range of contexts. Future research could also focus on real-world deployment scenarios, user experience evaluations, and optimizing the system for resource-constrained devices. These directions pave the way for continued advancements in the field, fostering innovation and addressing emerging challenges.

REFERENCES

[1] "Generative Adversarial Text-to-Image Synthesis" by Reed et al.: This work introduced a method using a combination of generative adversarial networks (GANs) and recurrent neural networks (RNNs) to generate realistic images from textual descriptions, 2016.
[2] Han Zhang[1], Tao Xu[2], Hongsheng Li[3], Shaoting Zhang[4], Xiaogang Wang[5], Xiaolei Huang[6], Dimitris Metaxas[7]: StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks, 2017.
[3] Zizhao Zhang[*], Yuanpu Xie[*], Lin Yang University of Florida: Photographic Text-to-Image Synthesis with a Hierarchically-nested Adversarial Network, 2018.
[4] "AttnGAN: Fine-Grained Text to Image Generation with Attention Generative Adversarial Networks" by Xu et al.: This work introduced an attention mechanism to the text-to- image synthesis process, allowing the model to attend to different parts of the text during image generation, 2018.
[5] "MirrorGAN: Learning Text-to-image Generation by redescription" by Chen et al.: The authors presented a redescription-based framework for text-to-image synthesis, which involves generating image descriptions that are then used to reconstruct the corresponding images, 2019.
[6] Pranjal Jain[1], Tanmay Jayaswal[2]: Department of Computer Science and Technology, SRM Institute of Science and Technology, Chennai, Tamil Nadu, India, 2020.
[7] Stanislav Frolova,b , Tobias Hinzc , Federico Raueb , J¨orn Heesb , Andreas Dengela: Adversarial Text-to-Image Synthesis, 2021.
[8] Rihito Tominaga[*] and Masataka Seo[*:] Image Generation from Text Using StackGAN with Improved Conditional Consistency Regularization, 2022.
[9] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with Latent Diffusion Models,” arXiv.org, 13-Apr-2022. [Online]. Available: https://arxiv.org/abs/2112.10752.
[10] J. Alammar, “The Illustrated Stable Diffusion,” The Illustrated Stable Diffusion — Jay Alammar — Visualizing machine learning one concept at a time. [Online].Available: https://jalammar.github.io/illustrated-stable-diffusion/.
[11] A. Gordić, “Stable diffusion: High-resolution image synthesis with latent diffusion models | ML coding series,” *YouTube*, 01-Sep-2022. [Online]. Available: https://www.youtube.com/watch?v=f6PtJKdey8E.
[12] M. Arjovsky and L. Bottou. Towards principled methods for training generative adversarial networks. In
[13] ICLR, 2017.
[14] A. Brock, T. Lim, J. M. Ritchie, and N.Weston. Neural photo editing with introspective adversarial networks. In ICLR, 2017.
[15] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li. Mode regularized generative adversarial networks. In ICLR, 2017.
[16] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In NIPS, 2016.
[17] E. L. Denton, S. Chintala, A. Szlam, and R. Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In NIPS, 2015.
[18] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In ICML, 2016.
[19] A. van den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu. Conditional image generation with pixelcnn decoders. In NIPS, 2016.

