I. J. Engineering and Manufacturing, **2024, 1, 24-37**
Published Online on February 8, 2024 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijem.2024.01.03

Generation of Images from Text Using AI

Nimesh Yadav*
Student, Department of Computer Engineering, School of Technology Management and Engineering, NMIMS University, Mumbai, India
Email: nimesh.yadav72@nmims.edu.in
ORCID iD: https://orcid.org/0000-0001-7807-1667
*Corresponding Author

Aryan Sinha
Student, Department of Computer Engineering, School of Technology Management and Engineering, NMIMS University, Mumbai, India
Email: aryan.sinha67@nmims.edu.in
ORCID iD: https://orcid.org/0000-0001-9348-6641

Mohit Jain
Student, Department of Computer Engineering, School of Technology Management and Engineering, NMIMS University, Mumbai, India
Email: mohit.jain25@nmims.edu.in
ORCID iD: https://orcid.org/0009-0007-1629-6677

Aman Agrawal
Student, Department of Computer Engineering, School of Technology Management and Engineering, NMIMS University, Mumbai, India
Email: aman.agrawal04@nmims.edu.in
ORCID iD: https://orcid.org/0009-0006-0881-4158

Sofia Francis
Faculty, Department of Computer Engineering, School of Technology Management and Engineering, NMIMS University, Mumbai, India
Email: Sofia.Francis@nmims.edu
ORCID iD: https://orcid.org/0000 0002 2339 8029

Received: 12 March, 2023; Revised: 11 June, 2023; Accepted: 23 July, 2023; Published: 08 February, 2024

Abstract: Reading the words can be confusing, and it may be hard to picture what is happening. There are some circumstances where words can be misunderstood. It's much simpler to recognize text if it's displayed as an image. The use of visuals is proven to increase viewership and retention.

Synthesizing realistic images automatically is a challenging undertaking, and even the most advanced artificial intelligence and machine learning algorithm has trouble meeting this standard. GANs (Generative Adversarial Networks) are just one example of a powerful neural network architecture that has shown promising results in recent years. Existing text-to-image methods can generate examples that generally reflect the meaning of the provided descriptions, but they often lack the necessary details and colorful object elements.

The primary objective of our research was to explore diverse architectural methodologies with the intention of facilitating the generation of visual representations from textual descriptions. By delving into this investigation, we aimed to discover and examine various approaches that could effectively support the creation of visuals that accurately depict the content and context provided within written narratives. Our aim was to unlock new possibilities in the realm of visual storytelling by establishing a strong connection between language and imagery through innovative architectural techniques.

Index Terms: Image generation, GAN, text to image, Artificial intelligence, Machine learning.

1. Introduction

Generating realistic images based on textual descriptions is a significant advancement in enhancing user intelligence. Visual mental imagery, or "seeing with the mind's eye," plays a crucial role in various cognitive processes like learning, memory retention, and logical reasoning. The ability to create a system that comprehends the relationship between sight and language and can produce images that convey the meaning of textual descriptions, holds great potential for revolutionizing multiple industries, including interactive computational graphic design, image fine-tuning, and animation. However, achieving photorealistic images from textual descriptions has proven challenging for many advanced approaches, particularly due to the multimodal nature of plausible images corresponding to a given text description. Generative models, especially Generative Adversarial Nets (GANs), have shown promising progress in generating realistic images. T2I (text-to-image) and I2T (image-to-text) generation are emerging areas of research where models can create photos from text or generate textual descriptions based on images.

The application of GANs in the layout of generative models has further improved their performance in capturing and recreating diverse content from existing data. As a result, GAN models have gained widespread adoption in recent years. This progress in generating images from text descriptions holds immense potential and has made T2I generation an essential area of study across various domains. This paper aims to develop a text-to-image generation model using our novel architecture named GAN-CLS, which stands for Generative Adversarial Networks - Conditional Latent Space. By combining the CLS algorithm with GAN, we can produce images that outperform those generated solely by the GAN algorithm. The primary focus of our research is to demonstrate the superior results achieved through this innovative approach in generating images from text descriptions.

Example:

Text description: In contrast to its bright yellow stamen, this white flower's petals are delicate, thin and dainty.

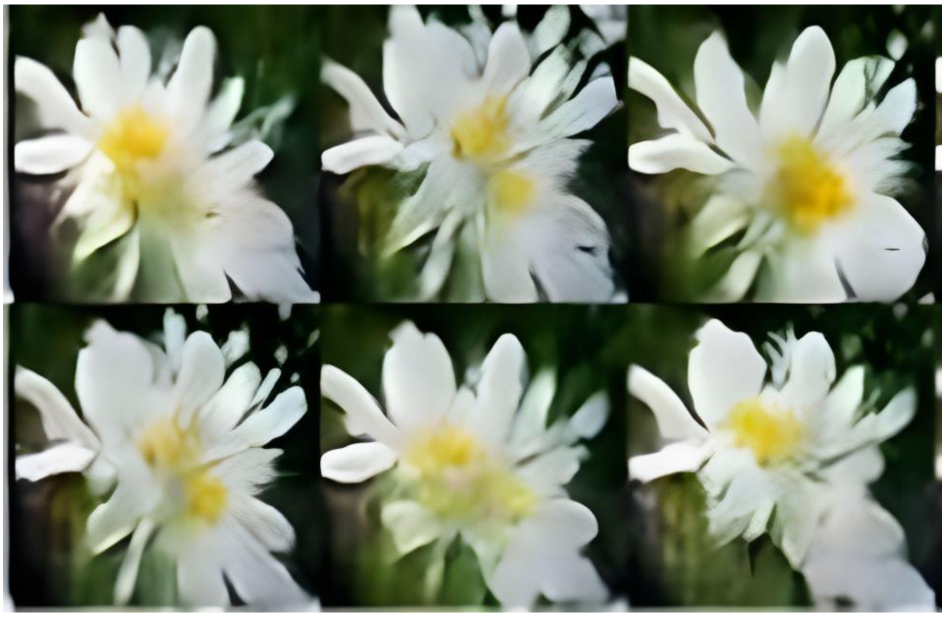


Fig.1. Diagram of Expected Generated Image

Thus, through this research we have attempted to talk in depth about image generation from text and the various key factors and how our research was successful with it. The literature study is described in Section 2 of this paper, and the analysis and design of the approach we suggest are covered in Section 3. Section 4 discusses the process for creating visuals from text, and Section 5 discusses the outcomes of our study. Section 6 contains the conclusion. The references and citations are discussed in Section 7.

2. Literature Review

The process of translating textual prompts into visually coherent and lifelike images has witnessed significant progress in recent times, primarily driven by the rapid evolution of deep learning models.

Our comprehensive survey encompasses a broad spectrum of studies, ranging from early approaches employing conventional image synthesis techniques to the more sophisticated and efficient AI-powered frameworks, such as Generative Adversarial Networks (GANs) and Transformer-based architectures. Through a meticulous analysis of the strengths and limitations inherent in each approach, we aim to offer profound insights into the current landscape of techniques, thereby identifying potential areas that warrant further advancements.

This literature survey constitutes indispensable bedrock for our research pursuits in crafting an innovative image generation model that extends the frontiers of AI's capacity to translate textual descriptions into visually captivating and exceptionally realistic images.

Table.1. Table of literature review and survey

Citation	Methodology	Architecture	Limitations
[2]	Used Pathways Autoregressive Text-to-Image (Parti) model	To begin with, Parti encodes images as collections of distinct tokens using the Transformer-based image tokenizer ViT-VQGAN. Second, the encoder-decoder Transformer model is scaled up to 20B parameters, with a new, cutting-edge zero-shot FID score of 7.23 and finely tuned FID score of 3.22 on MS-COCO.	1) Color bleeding 2) Feature bending 3) Hallucination or duplication of details
[13]	(VQ-Diffusion) model for text-to-image generation. This method is based on a vector quantized variational autoencoder (VQ-VAE) whose latent space is modeled by a conditional variant of the recently developed Denoising Diffusion Probabilistic Model (DDPM).	In this paper a transformer that encodes and decodes data was suggested to estimate the distribution $p\theta(\tilde{x}0|xt, y)$. The framework consists of two components: a diffusion picture decoder and a text encoder. The text encoder produces a conditional feature sequence from the text tokens y. The noiseless token distribution $p\theta(\tilde{x}0|xt, y)$ is produced by the diffusion image decoder using the image token xt and the timestep t. A softmax layer and many transformer blocks are included in the decoder.	Still have weaknesses of unidirectional bias and accumulated prediction errors due to the limitation of AR models
[14]	Proposed the Lafite first work to train text-to-image generation models without any text data. Our method leverages the well-aligned multimodal semantic space of the powerful pre-trained CLIP model: By creating text features from image features, the need for text-conditioning is seamlessly reduced.	For this purpose, we propose converting the unconditional StyleGAN2 to a conditional generative model. Because the proposed LAFITE is a flexible system, we run tests in a variety of configurations, including the suggested language-free option, as well as the zero-shot and fully-supervised text-to-image creation settings.	1) Color bleeding 2) Feature bending
[15]	Proposed Visually Guided Language Attention GAN (LatteGAN), a multi-turn text-conditioned image generation GAN accompanied by two key components: a Latte module that can extract the fine grained instruction representations that are crucial for image modification; and a Text-Conditioned U-Net discriminator that can discriminate images on the basis of both their modification and their quality.	They introduce an innovative architecture referred to as a Visually Guided Language Attention GAN (LatteGAN). In this paper, the authors address the limitations of previous approaches by introducing the Visually Guided Language Attention (Latte) module, which extracts fine-grained text representations for the generator, and the Text-Conditioned U-Net discriminator architecture, which discriminates both the global and local representations of fake and real images. Extensive trials on two separate MTIM datasets, CoDraw and i-CLEVR, reveal that the proposed model achieves state-of-the-art performance.	The current methods often overlook manipulation instructions and fail to generate objects. This is particularly problematic for the MTIM task, as generating an image at a certain step often involves reference to the previous images.
[21]	Suggested a new generative adversarial network (ManiGAN) with the text-image affine combination module (ACM) and the detail correction module (DCM) as its two main components.	Choose the multi-stage ControlGAN architecture as the fundamental structure since it generates high-quality and controllable images based on the provided text descriptions. In order to extract regional picture representations, we additionally incorporate an image encoder that is a pretrained Inception-v3 network.	1) Spatial relations 2) Incorrect visual aspect and media blending
[22]	Used NUWA-Infinity, a generative model for infinite visual synthesis, which is defined as the task of generating arbitrarily-sized high-resolution images or long-duration videos.	According to the paper to handle such a variable-size generation task, an autoregressive over autoregressive generation mechanism is proposed, where a global patch-level autoregressive model takes into account dependencies between patches and a local token-level autoregressive model takes into account dependencies between visual tokens within each patch.	-
[33]	Unified multimodal pretrained model called NUWA that can generate new ¨or manipulate existing visual data (i.e., images and videos) for various visual synthesis tasks.	A 3D encoder-decoder framework is created that can not only handle movies as 3D data, but also words and pictures as 1D and 2D data, respectively. A 3D Nearby Attention(3DNA) mechanism is also proposed to consider the nature of the visual data and reduce the computational complexity.	They treat images and videos separately and focus on generating either of them. This limits the models to benefit from both image and video data

3. Analysis and Design

The proposed architecture diagram is as per the following hardware and software specifications:
Hardware Specification:

- Intel processor i5 and above
- 8 GB RAM
- 500 GB hard disk

Software Requirements:

- Visual Studio Code
- Python 3.6
- Google Colab

3.1 Generative Models

When it comes to statistics, all Generative models belong to the same category because of their ability to produce novel data samples. To execute tasks like probability and likelihood estimation, modeling data points to characterize the phenomena in data, and distinguishing between classes based on these probabilities, unsupervised machine learning makes use of these models. Generative models are well-suited to the process of text to image synthesis since it describes the problem they are trying to address. Generative models are able to take on more difficult problems than their discriminative counterparts since they usually use the Bayes theorem to establish the joint probability [1]. Learning algorithms aim to imitate the underlying patterns or distribution of the data points, while generative models focus on the distribution of specific classes within a dataset. These models produce examples in which a specified input (x) and the desired output (y) occur simultaneously using the concept of joint probability.

Let's say we have m samples in a dataset $X = \{x^{(1)}, \ldots, x^{(m)}\}$ and each sample's $x^{(i)}$ is a vector. Here, the image will be represented as a vector of pixel values, denoted by $x^{(i)}$.

The dataset's images are drawn at random from a distribution termed P_r (where r stands for "real"), which has not been formally defined. The purpose of these generative models is to learn to generate samples from a distribution P_g that estimates a target distribution $P_r . P_g$, the model distribution, is a hypothesis about the true P_r, the data distribution.

To learn a distribution P_g, most generative models optimize the expected log-likelihood $E_{X\sim P} rlog(P_g(x|\theta))$ with regard to θ. Simply put, maximum probability learning corresponds intuitively to giving more weight to parts of X that have more examples from X and less weight to regions that have fewer examples. It may be shown that minimizing P_g is equivalent to maximizing the Log-likelihood. Assuming P_r and P_g are densities, the Kullback-Leibler divergence

$$KL(P_r \mid\mid P_g) = \int \quad XP_r log(\frac{P_r}{P_g})dx.$$

The expectation may be roughly estimated with enough samples in accordance with the weak law of large numbers, making this method appealing in part because it eliminates the requirement to know the unknown P_r. Genetic adversarial networks (GANs) are another sort of model that uses a strategy inspired by game theory [2].

3.2 Generative Adversarial Networks (GAN)

Generative Adversarial Networks (GANs) are a method for unsupervised learning that creates new instances. New data examples are generated with the help of neural networks in Generative Adversarial Networks [3]. It has the potential to provide both visual and aural content. Learn a generative model and train it with neural networks; this is the essence of generative adversarial networks.

In GAN, a discriminator and a generator, both models of neural organization, are put up to compete with one another in order to notice, catch, and replicate the diversity present in a dataset. In text to image the description of an image based on textual information is synthesized into a visual representation that best fits the description.

Most problems with existing generative models are addressed by GANs:

- Images created with GANs are of higher quality than those created with any of the other models.
- As will be explained below, GANs are not limited by the possibility that there is no such thing as a density P_g for complicated distributions and hence no need to learn such a thing.
- A GAN may produce samples quickly and in parallel. It's possible, for instance, to create an image simultaneously, rather than one pixel at a time.
- Loss functions and sample-generating network architecture can be modified in GANs with relative ease.

- $P_g = P_r$ is true when GANs converge. Other models, which may also optimize a loss, may have a biased estimator, therefore this equality would not apply to them.

The GAN framework revolves around a min-max game between two players, the discriminator (or critic) and the generator.

a. Generator

It creates new data instances, most of which are fabricated, and sends them on to the Discriminator in an effort to throw it off its scent.

b. Discriminator

As recently referenced, it can tell the difference between authentic samples and sham ones produced by the Generator. Each of the two neural networks, In both the Generator and the Discriminator, we find deep neural networks at work [4] . While the Discriminator is looking for truthful information, the Generator is trying to trick it. The Discriminator and the Generator have a hostile relationship with one another. The Generator does everything it can to fool the Discriminator into thinking the fake photo instances are the real samples of data, while the Discriminator is responsible for actually identifying the real ones. For this reason, these procedures are performed frequently until both sub-models are adequately trained. To ensure the discriminator can accurately recognize authentic data, it is first trained on a set of test samples. To test the Discriminator's ability to tell the difference between real and fake images, it is trained on synthetic data.

To further advance, Generator is additionally trained based on Discriminator's output. Generative Adversarial Network (GAN) has seen widespread use, and its extension, Deep Convolutional GAN, has seen even more success. Here, Generator must produce a vector, where vectors are composed of latent variables, in order to generate new data. During self-training, the GAN model consumes a substantial amount of time [5].

3.3 Conditional GANS

GANs can be easily converted into a conditional generative model, as detailed in the original GAN study [6]. The conditional generative model is a simple extension of the original GAN work, which outlines how to turn GANs into them. Generating information dependent on a given condition vector, the vector is attached to all layers of the generator and discriminator. After some time, the networks will figure out how to handle the new information and modify their settings accordingly.

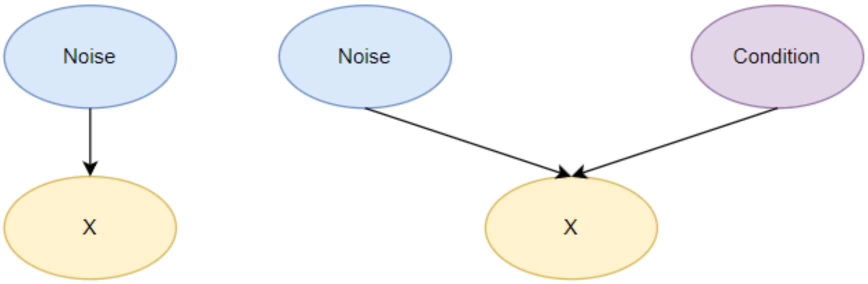


Fig.2. View of conventional GANs in a probabilistic graphical model (left) and a conditional GAN (right).

One can also see conditional GANs as probabilistic graphical models (Fig 2). The observable X is affected by the noise in typical GANs. Noise and Condition are both considered while determining X in conditional GANs. Condition vectors are vectors encoding text description in the context of text-to-image synthesis.

3.4 Proposed Architecture GAN-CLS

Artificial intelligence has significant challenges when attempting to translate visual output from textual format. The use of automatic picture synthesis has numerous advantages [7]. One use case for conditional generative models is image generation. The use of Generative Adversarial Networks has led to recent advancements (GAN). The text-to-image transformation is a perfect illustration of the power of deep learning. Text-to-image synthesis poses a significant challenge to our capacity to characterize conditional, high-dimensional distributions, but it also has many exciting and useful applications, such as photo editing and computer-assisted content development. In addition, we present distance-based conditional image creation using CLS, a new model that we believe provides more reliable stability guarantees than previous approaches (Fig 3).

The steps necessary to teach a GAN to respond to conditions are, it's easiest to treat (text, picture) pairs as a combined observation, and instruct the classifier to determine whether or not they are genuine. The discriminator does not know for sure if the text embedding context corresponds to the actual training images. In GAN-CLS, a third type of input is provided during training consisting of authentic images with manipulated text that the classifier must learn to identify as spurious. Training the discriminator to optimize picture/text matching in addition to image realism improves its ability to provide a signal to the generator.

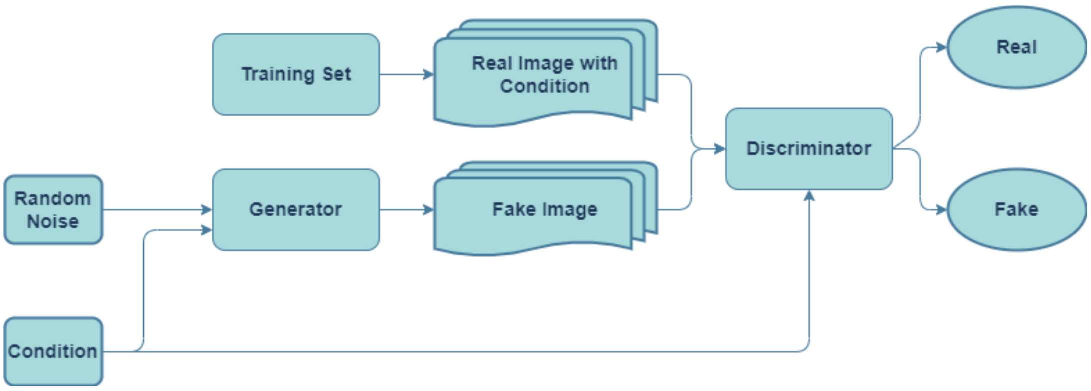


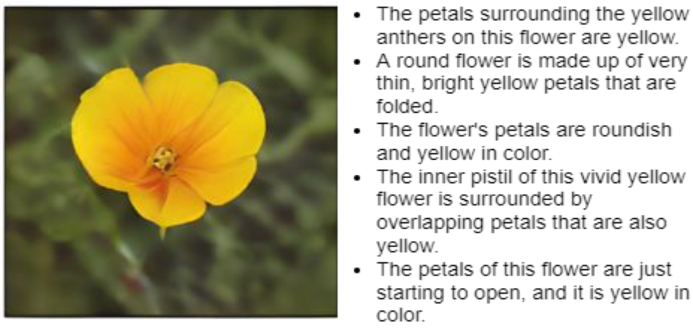
Fig.3. Proposed Model Architecture GAN-CLS

4. Methodology

This section discusses the proposed novel architecture using GAN-CLS model and adding word embedding using glove embeddings of text in the preprocessing phase and then training our model [8].

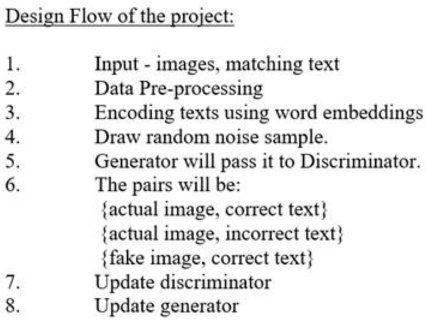
Incorporating the CLS algorithm proved to be a pivotal enhancement in our GAN model, yielding substantial improvements in the generated results. By integrating the CLS algorithm, we introduced a carefully crafted noise injection mechanism that effectively refined the discriminator's judgment process. This noise injection facilitated the discriminator in making more informed and precise assessments of the generated images. Consequently, the strengthened capabilities of the discriminator induced the generator to produce superior results, further compelling the discriminator to deceive with increasingly high-quality image outputs. As a result of this synergistic interaction, our GAN model successfully achieved a significant advancement in generating exceptionally realistic and visually appealing images.

The datasets used in this model are publicly available datasets. In this study we have used the Oxford-102 flowers dataset [24]. This dataset is the one which is typically employed in studies involving the conversion of text to visuals. The 102 flower types represented in Oxford-102 total 8,192 photographs. There are pictures, but no accompanying text, in this dataset. Nonetheless, we made advantage of the publicly available Amazon Mechanical Turk-collected captions for these datasets. Five different explanations are provided for each of the photographs. They have ten words or more, they do not give any context, and do not identify the flower depicted.



- The petals surrounding the yellow anthers on this flower are yellow.
- A round flower is made up of very thin, bright yellow petals that are folded.
- The flower's petals are roundish and yellow in color.
- The inner pistil of this vivid yellow flower is surrounded by overlapping petals that are also yellow.
- The petals of this flower are just starting to open, and it is yellow in color.

Fig.4. The figure shows an image with its corresponding captions on the right side.



Design Flow of the project:

1. Input - images, matching text
2. Data Pre-processing
3. Encoding texts using word embeddings
4. Draw random noise sample.
5. Generator will pass it to Discriminator.
6. The pairs will be:
   {actual image, correct text}
   {actual image, incorrect text}
   {fake image, correct text}
7. Update discriminator
8. Update generator

Fig.5. Proposed Algorithm.

4.1 Steps of Algorithm

a. Data Pre-processing:

The data must be prepared for the model in this stage. To make the photos easier to deal with in the deep learning model, they are transformed into NumPy arrays with pre-set pixel sizes [9]. Using GloVe, an unsupervised learning technique that creates vector representations of words, the picture descriptions are also transformed into embeddings (numerical vectors). The captions are kept in a CSV file that will subsequently be used to match captions to created photos. Lastly, GloVe embeddings are used to encapsulate both matching and mismatching text descriptions.

b. Loading and Combining

The model is given the pre-processed picture and caption data. While the picture description embeddings are loaded individually, the image data is concatenated to create a single NumPy array.

c. Data Modelling

At this stage, the model architecture is defined. It entails building the generator and discriminator networks. The generator network produces a produced picture that fits the input description after receiving random noise and a written description as inputs. A probability indicating whether or not the picture fits the description is produced by the discriminator network after receiving an image and a text description as input. In an adversarial training process, the discriminator and generator networks work together to develop their respective capabilities [10]. As a result, the discriminator becomes more adept at telling the difference between real and generated images, and the generator becomes more realistic in the images it produces. This stage also defines the loss functions for the generator and discriminator networks, which are used to gauge the effectiveness of the networks during training.

d. Training

This stage involves training the model. The train step function uses the generator network to create pictures, estimates the loss for both the discriminator and generator networks, and modifies the gradients to enhance the performance of the network. The train function gathers all the metrics, including the loss and accuracy for each epoch, and then collects the batch data, passing it to the train step function.

e. Results

A function is used to test the output from the generator once the model has been trained. This function creates a picture that fits the provided description by taking two inputs—random noise and a caption. A created image that can be assessed for quality and contrasted with the supplied caption is the output of the function.

4.2 About the GAN Network

It specifies that three sets of inputs will be given to the Discriminator as a generator creates bogus samples and sends them on [11]. The most precise result is the combination of correct text and actual image, followed by the incorrect or fake text and actual image, and finally the false image with proper text. The Discriminator is trained using these inputs to improve its performance.

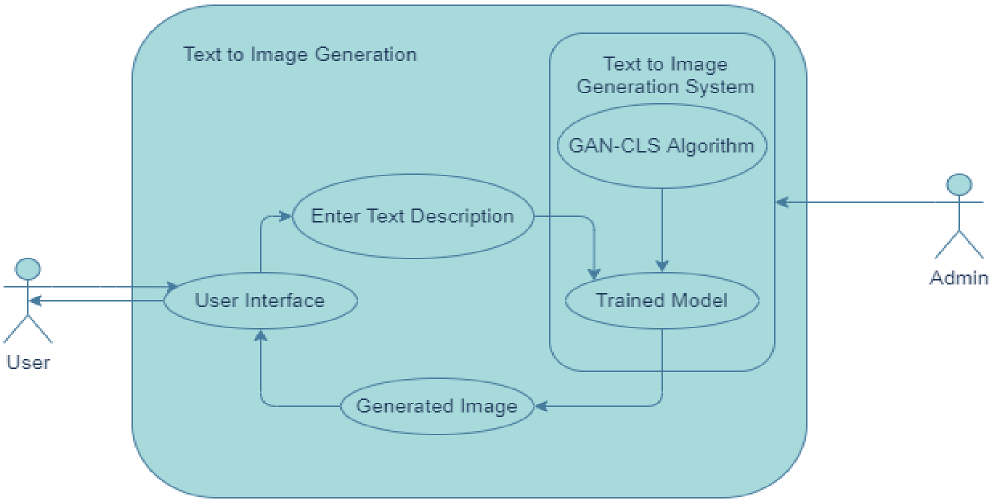


Fig. 6. Use Case Diagram of the proposed model.

4.3 Limitations and their solutions

Our data is limited in size by a lack of images; therefore, we perform data augmenting on them. We randomly crop the images and perform left-right-flipping on them. To ensure that the two sets of data are sufficiently distinct, we created a separate set. It is recommended to create a "train" set and a "test" set with different types of images in each.

5. Results

After the training process of our model, we run a sample test on our generator by providing it some noise and some captions from our dataset and we can see the result in Fig 7 below.

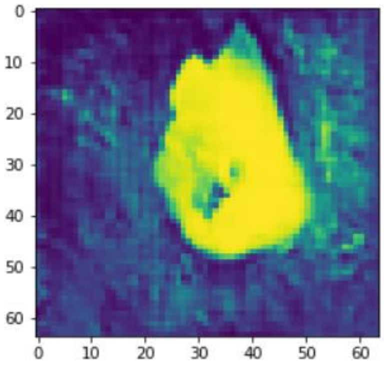


Fig. 7. Sample output from the Generator

Then we initialize our discriminator for evaluation purpose and a decision function is passed to the discriminator consisting of the sample image in Fig 8 and the corresponding word embeddings that were provided at the time of generation. The discriminator then makes its decision by loading pre-trained weights for the discriminator model that have been learned from previous training sessions and judges the generator’s performance.

In contrast to the improved performance of the generator after training, the discriminator's accuracy decreases because it becomes increasingly difficult for it to distinguish between genuine and fabricated data. For a perfect generator, the discriminator can only expect to be 50% accurate. Accuracy of our discriminator model has come up to 43.442073%.

Then we move to the testing phase of the project, where we take a sample input for image generation from the user and let the model generate a frame of 7x4(28) images which is saved and shown as output. Following are some examples from the testing phase of our model.

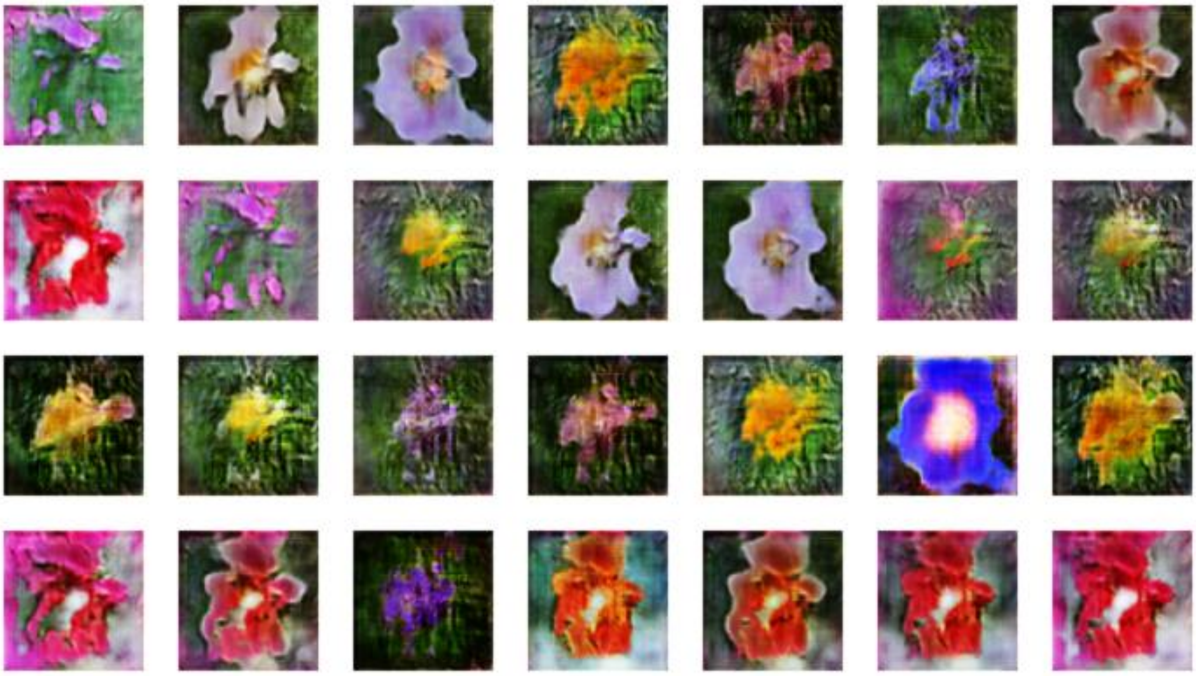


Fig.8. Caption: - “this flower is purple in color with oval shaped petals”

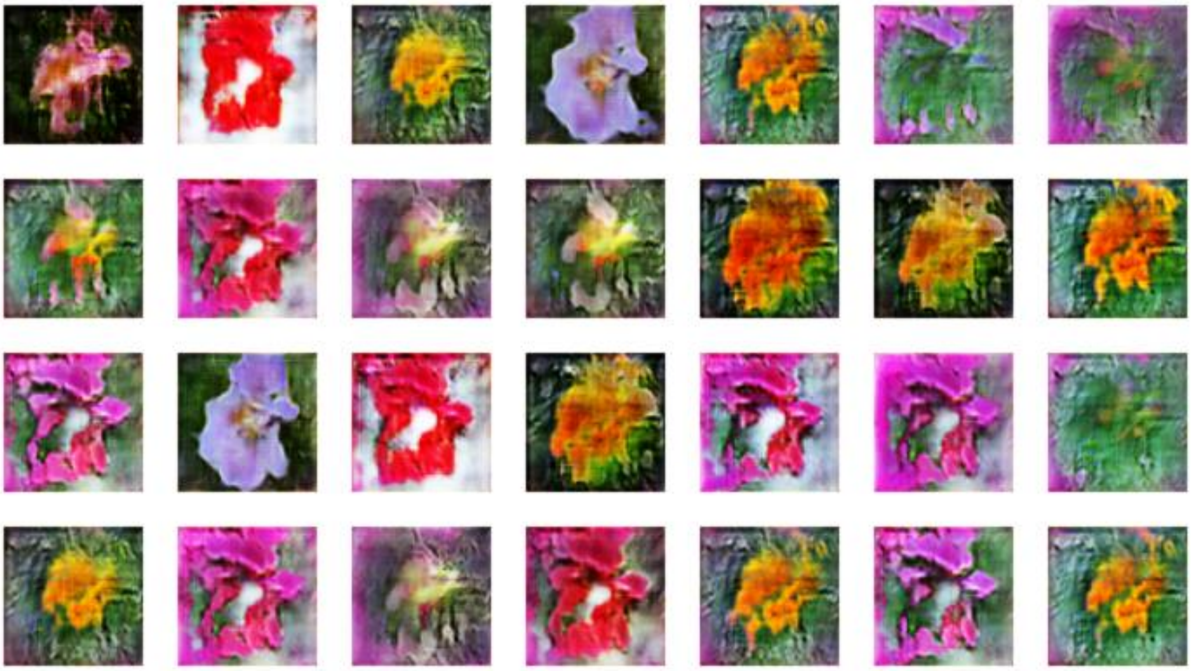


Fig.9. Caption: - “this flower is yellow in color with oval shaped petals”

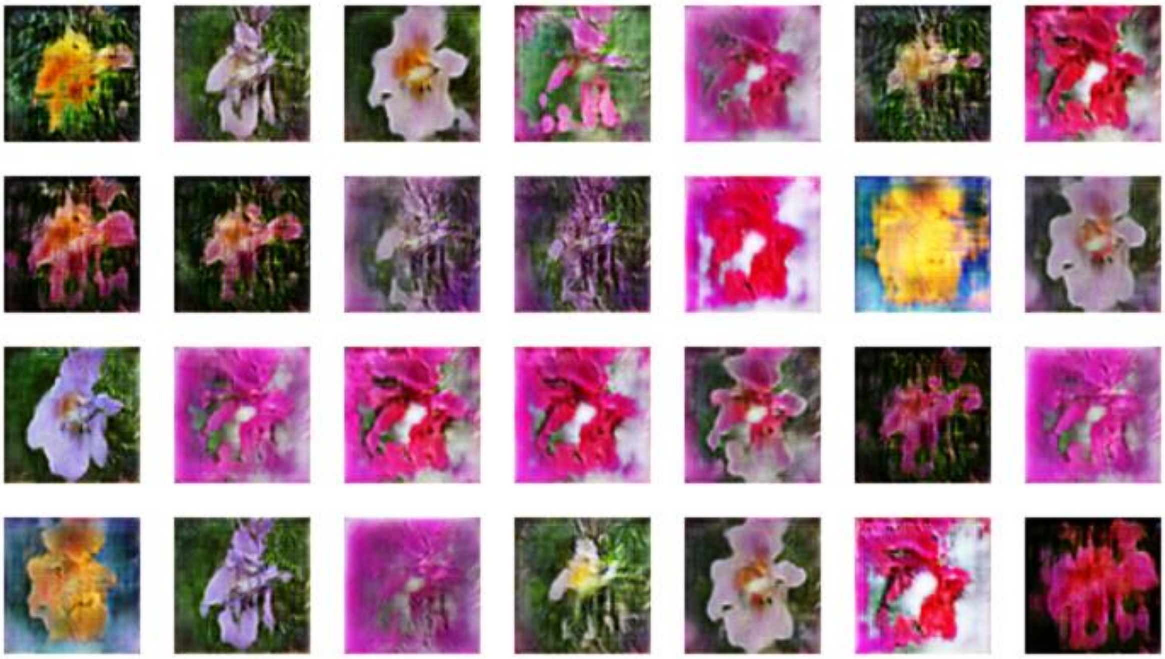
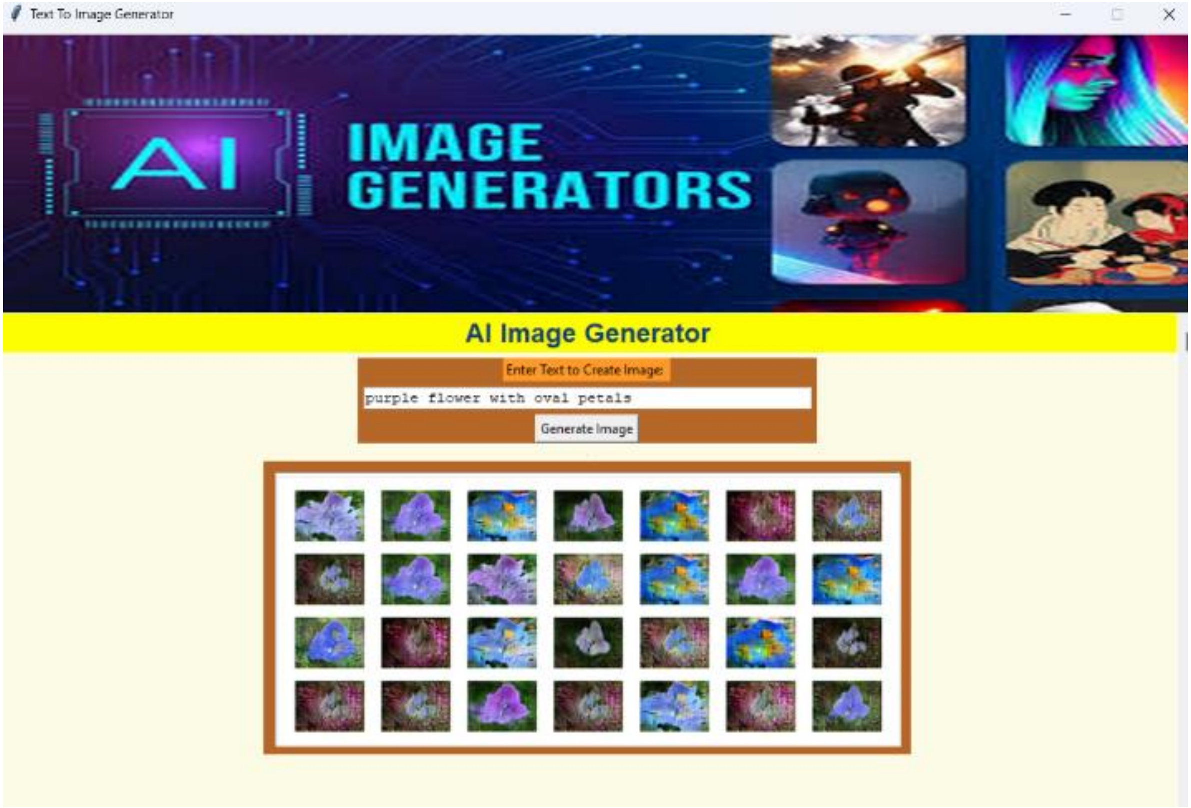


Fig.10. Caption: - “purple flower with curvy string-like petal sanda group of large yellow stamen in the center.”

Here, we have tested 3 captions on our system. We can clearly conclude that our GAN-CLS model was approximately able to generate 28 images of flowers which provide resemblance according to the provided input.

The proposed model is implemented in the form of a GUI dashboard having a user side and an admin side. The user will be able to interact with the GUI and enter the text that they wish to generate an image for. The text input from the user goes into the Text to Image Generation System which is basically our trained model along with the dataset and GAN-CLS algorithm. This image generation system will then return some images similar to the description provided by the user. These images in turn will be displayed to the user through the GUI application.

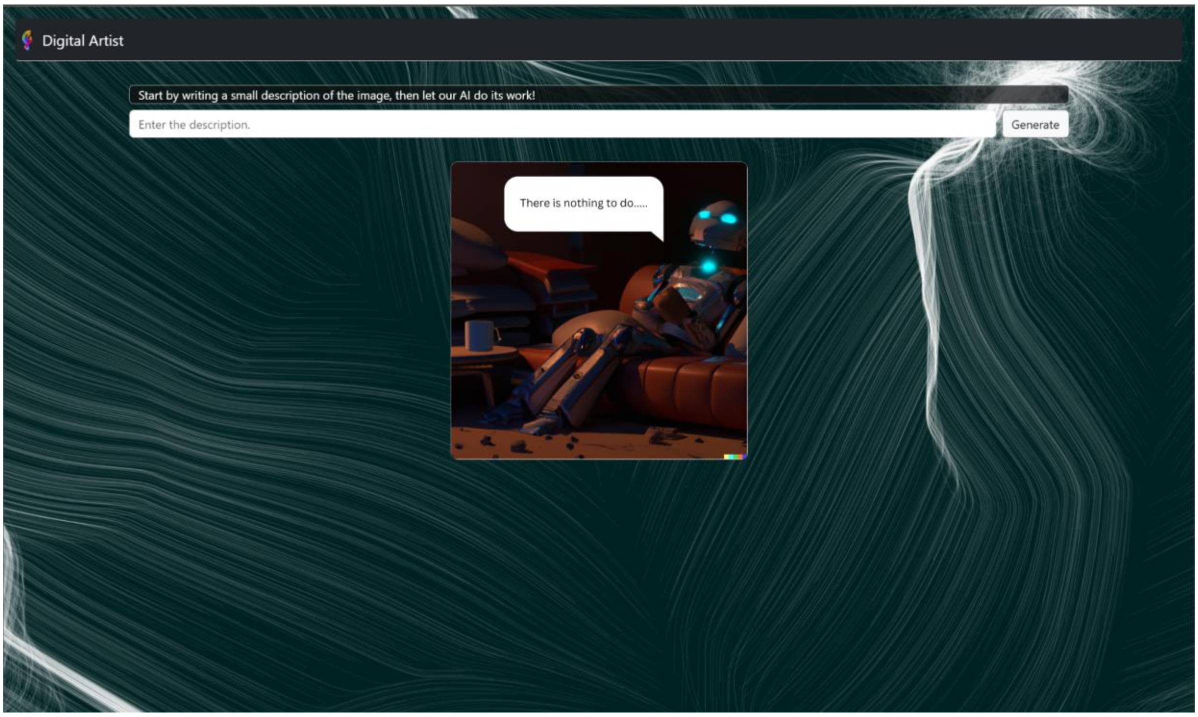


purple flower with oval petals

Fig.11. Sample GUI made with Tkinter

The above GUI which can be seen above in Fig 11 is a sample GUI screen made using Tkinter in Python to showcase a working prototype of our research.

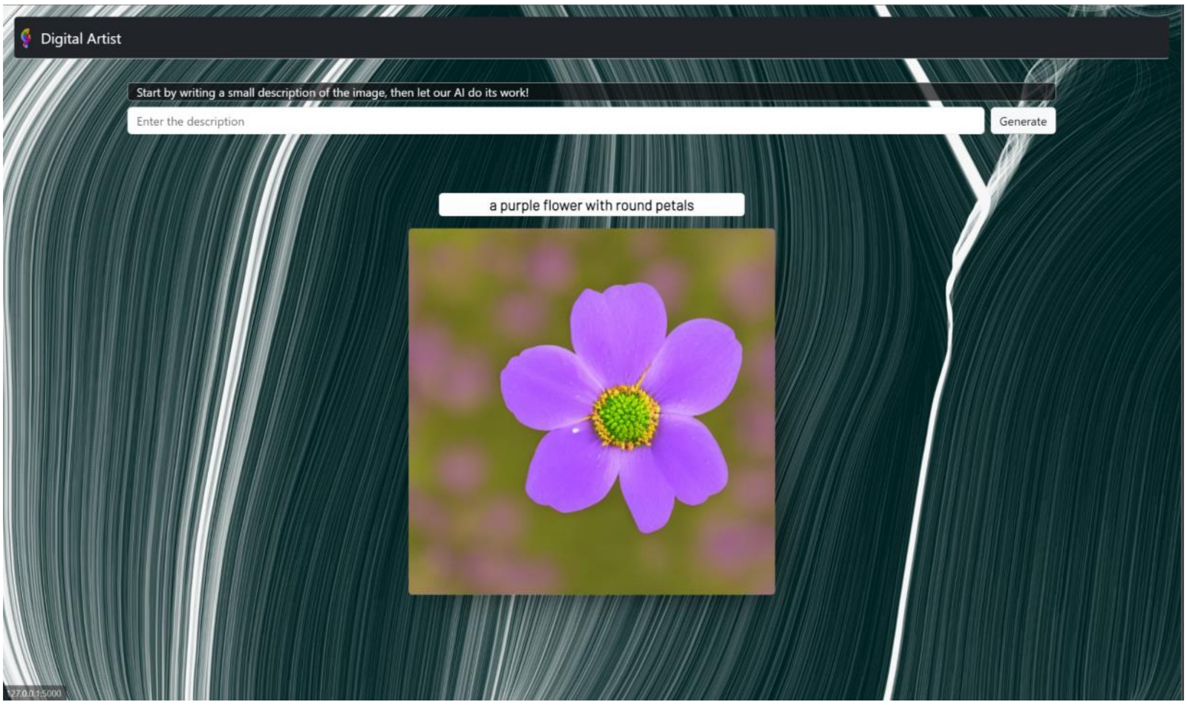
After the initial testing phase of the project, the focus shifted to the finalization phase of the project, where the interface was moved from Tkinter based GUI to Flask based website, which is visible in Fig.12.



Start by writing a small description of the image, then let our AI do its work!

Fig.12. Website made using Flask

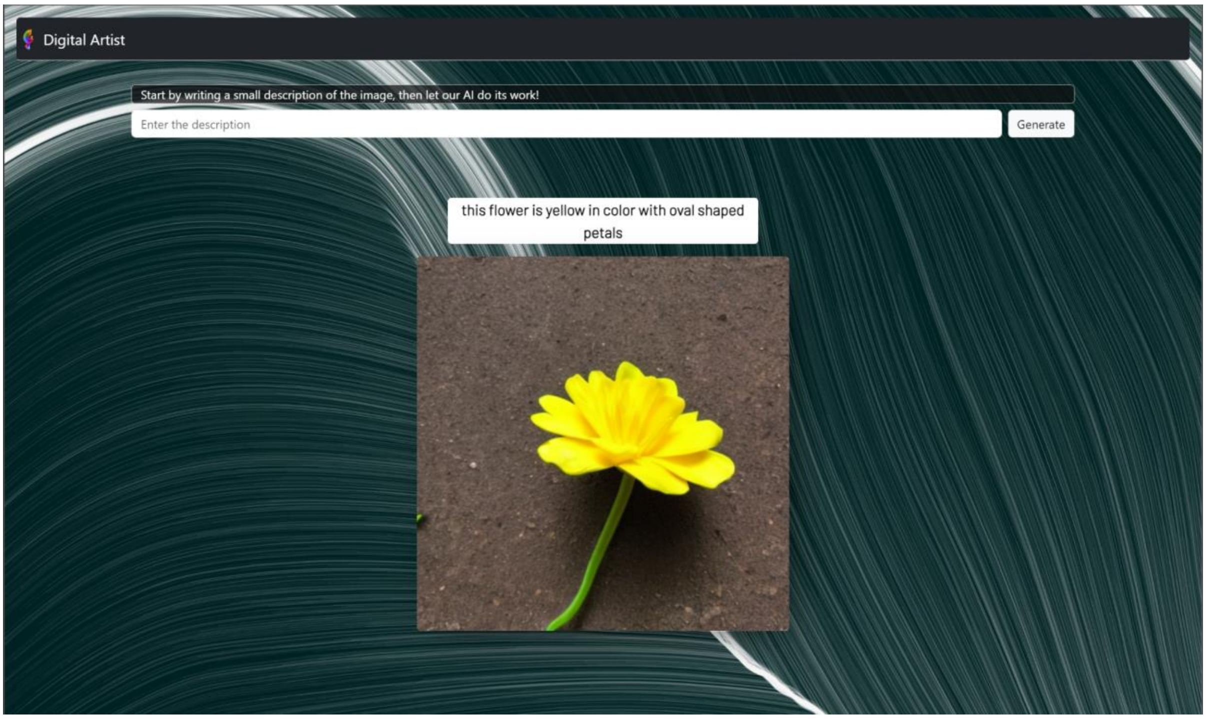
When the front interface of the system is finalized, the model is further improvised by increasing the epochs of the model and increasing the size of the dataset. The resolution of the image being generated is increased from 64px and instead of 28 images being generated in matrix of (7*4), in this iteration only 1 higher resolution image is being generated. In the below example Fig.13., we can see how the project has evolved by taking a sample input as “a purple flower with round petals”.


Start by writing a small description of the image, then let our AI do its work!

a purple flower with round petals

Fig.13. Sample output

We take another example with a previously tested caption during the testing phase as “this flower is yellow in color with oval shaped petals”.


Start by writing a small description of the image, then let our AI do its work!

this flower is yellow in color with oval shaped petals

Fig.14. Sample output

6. Conclusion

In this paper, we researched methodologies and structures with the aim of synthesizing pictures automatically from textual descriptions. We will implement improved architectures like the GAN-CLS to improve the performance. We introduce Generative Adversarial Networks and demonstrate its utility for the task of text to image synthesis in this paper. Here, we detail the state-of-the-art models in the field, which operate at the confluence of Computer Vision and Natural Language and explain how they achieve their impressive results. We also propose a new Conditional GAN (GAN-CLS) that permits conditional picture production via a dependable training process.

In summary, the goal of our study was to investigate several architectural techniques that may be utilised to bridge the gap between language and picture and produce visual representations from verbal descriptions. With the use of word embeddings during preprocessing and the CLS technique, we were able to produce realistic and visually attractive images with amazing improvements.

The incorporation of the CLS algorithm brought about a skillfully designed noise injection mechanism, enabling the discriminator to improve its evaluations of the generated images and make more educated decisions. The generator was then forced to generate outputs of higher quality as a result, creating a feedback loop that dramatically improved our GAN model's overall performance.

Acknowledgment

We may wish to honour everyone without whom our endeavour could not have been successful. Prof. Sofia Francis guided us throughout the project and provided tremendous assistance. Without her guidance, we would not have been able to realise how to correctly complete this research.

This work has helped us comprehend and communicate our theoretical knowledge in the practical world. So, we would really appreciate her support and leadership in this endeavour.

Also, we would like to thank ourselves, as every member of the team has consistently given their all and been available whenever required.

References

[1] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., ... & Sutskever, I. (2021, July). Zero-shot text-to-image generation. In International Conference on Machine Learning (pp. 8821-8831). PMLR.

[2] Yu, J., Xu, Y., Koh, J. Y., Luong, T., Baid, G., Wang, Z., ... & Wu, Y. (2022). Scaling autoregressive models for content-rich text-to-image generation. arXiv preprint arXiv:2206.10789..

[3] David Alvarez-Melis and Judith Amores. The emotional gan: Priming adversarial generation of art with emotion. In 2017 NeurIPS Machine Learning for Creativity and Design Workshop, 2017.

[4] Luca Bertinetto, F. Henriques, Jack Valmadre, Philip Torr, and Andrea Vedaldi. Learning Feed-forward one-shot learners. In D. D. Lee, M. Sugiyama, U. V. Luxburg,I. Guyon, and R.Garnett (eds.), Advances in Neural Information Processing Systems 29,pp. 523–531. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6068-learning-feed-forward-one-shot-learners.pdf.Oscar Chang, Lampros Flokas, and Hod Lipson. Principled weight initialization for hypernetworks.In International Conference on Learning Representations, 2020.

[5] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip H. S. Torr. 2019. Controllable text-to-image generation. Proceedings of the 33rd International Conference on Neural Information Processing Systems. Curran Associates Inc., Red Hook, NY, USA, Article 185, 2065–2075.

[6] Panos Achlioptas, Maks Ovsjanikov, Kilichbek Haydarov, Mohamed Elhoseiny, and Leonidas Guibas. Artemis: Affective language for visual art. arXiv preprint arXiv:2101.07396, 2021.

[7] Nilsback, Maria-Elena, and AndrewZisserman. ”Automated flower classification over a large number of classes.” Computer Vision, Graphics&ImageProcessing,2008.ICVGIP’08.SixthIndianConferenceon.IEEE,2008.

[8] Ivan Anokhin, Kirill Demochkin, Taras Khakhulin, Gleb Sterkin, Victor Lempitsky, and DenisKorzhenkov. Image generators with conditionally-independent pixel synthesis. arXiv preprintarXiv:2011.13775, 2020.

[9] Goodfellow,Ian,et al. ”Generative adversarial nets.” Advances in neural information processing systems.2014.

[10] Ahmed Elgammal, Bingchen Liu, Mohamed Elhoseiny, and Marian Mazzone. Can: Creative adversarial networks, generating” art” by learning about styles and deviating from style norms arXiv:1706.07068, 2017.

[11] Ahmed Elgammal, Bingchen Liu, Diana Kim, Mohamed Elhoseiny, and Marian Mazzone. The Shape of art history in the eyes of the machine. In Proceedings of the AAAI Conference onArtificial Intelligence, volume 32, 2018.

[12] Zhang, Han, et al. ”Stackgan:Text to photo-realistic image synthesis with stacked generative adversarial networks.”arXivpreprint(2017)

[13] Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., ... & Guo, B. (2022). Vector quantized diffusion model for text-to-image synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10696-10706).

[14] Zhou, Y., Zhang, R., Chen, C., Li, C., Tensmeyer, C., Yu, T., ... & Sun, T. (2022). Towards Language-Free Training for Text-to-Image Generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 17907-17917).

[15] Matsumori, S., Abe, Y., Shingyouchi, K., Sugiura, K., & Imai, M. (2021). LatteGAN: Visually Guided Language Attention for Multi-Turn Text-Conditioned Image Manipulation. IEEE Access, 9, 160521-160532.

[16] Reed, Scott, et al. ”Generative adversarial text to image synthesis.” arXivpreprintarXiv:1605.05396(2016).

[17] Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., & 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 (pp. 1316-1324).

[18] Nasr, A., Mutasim, R., & Imam, H. SemGAN: Text to Image Synthesis from Text Semantics using Attentional Generative Adversarial Networks. In 2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE) (pp. 1-6). IEEE.

[19] Özgen, A. C., Aghdam, O. A., & Ekenel, H. K. (2020, October). Text-to-Painting on a Large Variance Dataset with Sequential Generative Adversarial Networks. In 2020 28th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.

[20] Zhang,Han,et al.”Stackgan++: Realistic image synthesis with stacked generative adversarial networks. ”arXivpreprintarXiv: 1710.10916(2017).

[21] B. Li, X. Qi, T. Lukasiewicz and P. H. S. Torr, "ManiGAN: Text-Guided Image Manipulation," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 7877-7886, doi: 10.1109/CVPR42600.2020.00790.

[22] Wu. C, Liang. J, Hu.X, Gan. Z, Wang. J, Wangi. L, Liu. Z, Fang. Y and Duan. N, “NUWA-Infinity: Autoregressive over Autoregressive Generation for Infinite Visual Synthesis” 2022 arXiv.

[23] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, Honglak Lee , “Generative Adversarial Text to Image Synthesis" in University of Michigan and Max Planc Institute for Informatics June 2016.

[24] Yadav, N., & Sinha, A, “Augmented Reality and its Science”.

[25] Gregor, K., Danihelka, I., Graves, A., Rezende, D., and Wierstra, D. Draw: A recurrent neural network for image generation. In ICML, 2015.

[26] Ankit Yadav1, Dinesh Kumar Vishwakarma2, Recent Developments in Generative Adversarial Networks: A Review (Workshop Paper), 2020.

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

[28] Koh, J. Y., Baldridge, J., Lee, H., and Yang, Y. Text-toimage generation grounded by fine-grained user attention. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 237–246, 2021.

[29] Jain, M., Sinha, A., Agrawal, A., & Yadav, N. (2022, November). Cyber security: Current threats, challenges, and prevention methods. In 2022 International Conference on Advances in Computing, Communication and Materials (ICACCM) (pp. 1-9). IEEE.

[30] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. arXiv preprint arXiv:1606.03498, 2016.

[31] Mansimov, E., Parisotto, E., Ba, J. L., and Salakhutdinov, R. Generating images from captions with attention. arXiv preprint arXiv:1511.02793, 2015

[32] Tao, M., Tang, H., Wu, S., Sebe, N., Wu, F., and Jing, X.-Y. Df-gan: Deep fusion generative adversarial networks for text-to-image synthesis. arXiv preprint arXiv:2008.05865, 2020.

[33] Wu, Chenfei & Liang, Jian & Ji, Lei & Yang, Fan & Fang, Yuejian & Jiang, Daxin & Duan, Nan. (2022). NÜWA: Visual Synthesis Pre-training for Neural visUal World creAtion. 10.1007/978-3-031-19787-1_41.

Authors’ Profiles

Nimesh Yadav is a research scholar and currently pursuing dual degree course of Bachelor of Technology in Computer Engineering (CSE) and Master of Business Administration in Technology Management degrees from Mukesh Patel school of technology management and engineering, NMIMS University, Mumbai, India. He has worked as a research scholar and a Python Developer Intern in IBM and Phemesoft. He also has experience in the Finance sector as Treasury Analyst. His research interest includes research on applications of machine learning, deep learning, and artificial intelligence. He has also developed and created various large-scale projects on these topics.

Aryan Sinha is currently pursuing the Bachelor of Technology in Computer Engineering (CSE) and Master of Business Administration degrees from Mukesh Patel school of management and engineering, NMIMS University, Mumbai, India. His research interest includes research on applications of machine learning and cyber security. During the course of his academic journey, he demonstrated his skills and dedication by developing this project for the final year of his Bachelors in Computer Engineering.

Mohit Jain is a diligent individual currently pursuing a Bachelor of Technology in Computer Engineering (CSE) and concurrently undertaking a Master of Business Administration degree at Mukesh Patel School of Management and Engineering, NMIMS University, Mumbai, India. During the course of his academic journey, he demonstrated his skills and dedication by developing this project for the final year of his Bachelors in Computer Engineering. Mohit's research focuses on the practical applications of machine learning/deep learning, showcasing his passion for exploring innovative technologies and their implications.

Aman Agrawal is a hardworking person currently studying at Mukesh Patel School of Management and Engineering, NMIMS University, Mumbai, India. He is pursuing a Bachelor of Technology degree in Computer Engineering (CSE) and a Master of Business Administration degree. During his academic journey, Aman has shown his dedication and proficiency by creating a project in Computer Engineering for his final year. His research focuses on the practical use of machine learning/deep learning, demonstrating his enthusiasm to explore innovative technologies and their potential impact.

Prof. Sofia Francis is pursuing a PhD in Computer Engineering from Nirma University, Ahmedabad, India. She works as an Assistant Professor in the Department of Computer Engineering at NMIMS University, Mumbai, India. Her current research is in Deep Learning, Brain MRI analysis, Machine Learning, and pattern identification. She has done other notable research projects in Networking, too.

How to cite this paper: Nimesh Yadav, Aryan Sinha, Mohit Jain, Aman Agrawal, Sofia Francis, "Generation of Images from Text Using AI", International Journal of Engineering and Manufacturing (IJEM), Vol.14, No.1, pp. 24-37, 2024. DOI:10.5815/ijem.2024.01.03