IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 7, July 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

9940 572 462 | 6381 907 438 | ijircce@gmail.com | www.ijircce.com

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207044 |

Transforming Text into Visual Realities: Leveraging ST-GAN for Image Synthesis

Harshith Jain[1], Usha M[2]

MCA Student, Department of Computer Application, Bangalore Institute of Technology, Bangalore, India[1]

Assistant Professor, Department of Computer Application, Bangalore Institute of Technology, Bangalore, India[2]

ABSTRACT: The problem of generating high-quality images from text descriptions is a fascinating challenge in computer vision with vast potential applications like photo-editing and computer-aided design. although current AI systems still have limitations, substantial advancement has been achieved in text and image categories, paving the way for more advanced solutions. By leveraging the discriminative power classification accuracy and effective transfer learning of recurrent neural networks (RNNs) and convolutional neural networks (CNNs), these advancements can be harnessed to tackle this challenge. Generative Adversarial Networks (GANs) have shown promise in producing reasonably high-quality images of objects like birds and flowers. In this context, we propose a GAN architecture designed to effectively translate visual concepts from text descriptions into corresponding images. This approach aims to combine the strengths of RNNs and CNNs to bridge the gap between textual input and visual output, advancing the field toward more accurate and realistic image synthesis.

KEYWORDS: Generative Adversarial Networks (GANs), Generator, Discriminator, text to image, Residual GAN

I. INTRODUCTION

Among the top complex and fascinating challenges in Natural Language Processing (NLP) and Computer Vision is image captioning, where a system generates a narrative derived from a provided image. Conversely, text-to-image synthesis requires the system to provide a visual representation from a specified document, capturing the described visual properties. Creating images using vivid details and clarity from English text queries is a prime application of novel conditional generative models. Our objective within this piece is to create high-quality images from textual input. From an abstract perspective, this problem resembles language translation. Just as the same ideas can be expressed in different languages, images and textual content are visible as different languages encoding the same concepts. Complete knowledge pertaining to an entity can be represented through attribute representations that capture distinguishing characteristics of the object category, requiring the distinguishing capability and robust extrapolation properties of these representations. However, this poses a challenge since text-to-image or image-to-text conversions are highly multimodal, with many acceptable structures corresponding to one another (i.e., texts that correctly describe the image and vice versa). This difficulty can be mitigated by the sequential nature of language, allowing for the forecasting of the next word based on all previous words and the image.

The rationale for this work stems from recent advancements in Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), which have started to produce highly discriminative and generalizable text descriptions derived automatically from characters and words. This task can be decomposed into two main components: first, extracting important visual information from the text, where context is less helpful, making Word2Vec unsuitable since it cannot capture visual properties as effectively as an embedding specifically trained for this purpose. The second component involves using these learned features to generate images. The primary aim of our work is to develop a GAN architecture, specifically a Residual GAN (RGAN), to generate flower images with reasonable visual detail from given text descriptions.

II. DATASET USED

We used the publicly available Oxford-102 Flowers dataset, which consists of 8,192 images across 102 flower categories, with each category contributing between 40 and 258 images. This dataset contains only photos, without descriptions. Therefore, we utilized publicly available captions collected by Reed, which provide ten descriptions per image. Each description possesses a minimum of at least 10 words and does not specify the background or the flower species.

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207044 |

III. BACKGROUND

Text to image synthesis is based on GAN and in this section, we briefly describe the about the working of GAN [6].

A. Generative Adversarial Network (GAN)

The main idea behind GANs [6] involves two networks: a Generator network (G) that attempts to produce images, and a Discriminator network (D) that tries to distinguish between real and generated (fake) images. The Discriminator learns to map features extracted from the images to labels (real or fake) based on this correlation. Conversely, the Generator aims to create images exhibiting trades that are in accordance with a given label, rather than predicting a label from given characteristics.

GANs consist of G The adversary minimizes its loss when D(x)=1D(x) = 1D(x)=1 for real images and D(G(z)) =0 D(G(z)) = 0D(G(z))=0 for fake images, meaning it correctly identifies real images with a probability of 1 and fake images with a probability of 0. GANs can be conditioned on different variables, resulting in generated images that are influenced by these variables. Here, z represents the underlying "code" typically sampled from a simple distribution, such as a normal distribution. In a Conditional GAN, both the Generator and Discriminator are provided with extra conditioning factors ccc, leading to G (z, c) G (z, c)G(z,c) and D(x,c)D(x, c)D(x,c). This allows the Generator to produce images influenced by variables ccc, such as text descriptions.

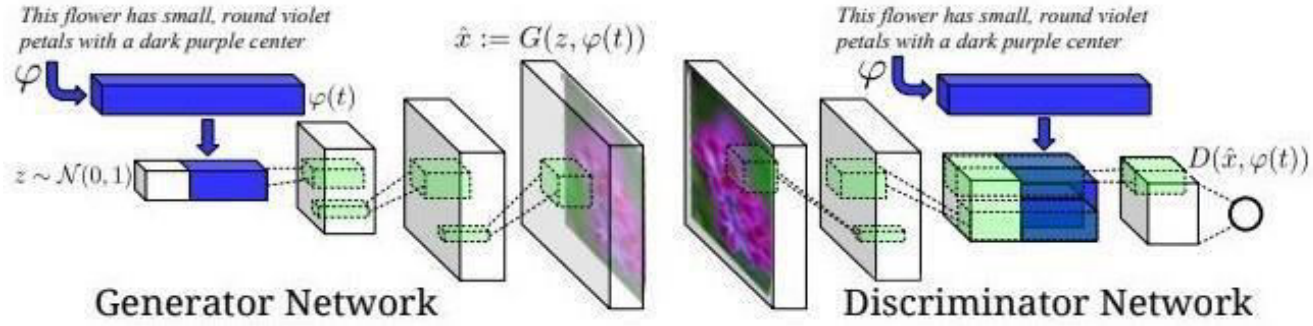

Fig. 1. Conditional GAN architecture. Text encoding ϕ(t) is used in both generator and discriminator [1]

and D indulging with each other in a two-player the opponent in technical jargon attempts to differentiate real images from counterfeit images, while the generator attempts to deceive the discriminator.

$$minG\ maxD\ V\ (D,tt)\ =\ Ex \sim p_{data(x)}[logD(x)] + Ex \sim p_{z\,(z)}[log(1\ -\ D(tt(z))] \quad (1)$$

Discriminator minimizes its loss when D(x) is equal to 1 and D(G(z)) is equal to 0, that is, when the discriminator's probability is 1 for real image and 0 for fake or synthesized image. While the generator attempts to maximize D(G(z)) meaning generator attempts to produce such type of images that discriminator thinks of as real images. It has proved profitable for the generator to optimize log(D(G(z))) instead of reducing the value of log (1-− D(G(z))).

GANs can be conditioned on different variables leading to the produced images conditioned on variables [1].

$$minwG\ maxwD\ V\ (D,tt)\ =\ Ex \sim p_{x(x)}[logD(x,wD)] + Ex \sim p_{z\,(z)}[log(1\ -\ D(tt(z,wG),wD)] \quad (2)$$

Where z is the underlying "code" that is usually sampled from an uncomplicated distribution (such as normal distribution). Conditional GAN is a form of GAN where both generator and discriminator gets extra conditioning variables c, producing G (z, c) and D (x, c). This form of GAN enables G to produce images dependent on variables c (here text).

IV. METHOD

To address the challenge of representing text in a visually discriminative format, we employ a Convolutional Neural Network (CNN) to obtain vector embeddings for images, and a Recurrent Neural Network (RNN) with Long Short-

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207044 |

Term Memory (LSTM) cells to convert text into vector form. Once we have these text and image embeddings, we define a loss function that aims to minimize the discrepancy between these embeddings based on their compatibility with
each other. The intuition behind our loss function that is "A text encoding should have a higher compatibility score with images of the corresponding class compared to any other class, and vice-versa" [1]. Specifically, we define the cost function using cosine similarity:
x_w = incorrect text embedding, v_w = incorrect image embedding, ξ= cosine similarity

$$Loss = max(0, \alpha - \xi(x, v) + \xi(x, vw)) + max(0, \alpha - \xi(x, v) + \xi(xw, v)) \quad (3)$$

The aim is to maximize the cosine similarity between the correct text and image embedding's, and simultaneously minimize it between incorrect pairs. This encourages the model to learn that the correct text embedding should closely match its corresponding image embedding. We propagate the resulting gradients through both the CNN and RNN components so that both networks learn to enhance the similarity between correct text and image pairs. In our implementation, we set the weighting parameter α\alphaα to 0.02 to balance the impact of this loss function during training.

V. RELATED WORK

Generative adversarial networks (GANs) introduced the adversarial learning framework and initially formulated a basic GAN structure and its training process [6]. They employed convolutional decoder networks for the generator module and discriminator. Since the inception of GANs in 2014, complex convolutional decoder architectures have been extensively utilized to generate realistic and vivid images. For example, deconvolution networks, comprising multiple conv2d and up sampling layers, were trained to create 3D chair renderings conforming to specific graphics rules governing shape, posture, and lighting conditions [11]. Efforts have concentrated on enhancing the clarity of produced images. However, these models typically weren't conditioned on external variables until Ryan Dahl's work on Pixel Recursive Super Resolution, which aimed to produce high-resolution images with sharp details conditioned on external variables—an ongoing area of research. Recent advancements have explored conditioning image generation on text descriptions rather than class labels to better capture visual details embedded in textual descriptions [5]. Our work stands out with specific architectural enhancements designed to improve overall model efficiency while achieving satisfactory results [9]. We also draw inspiration from related work on image restoration, particularly leveraging residual connections in both the generator and discriminator networks [10]. Furthermore, thanks to advancements in recurrent neural network decoders, these models have been utilized to generate written content based on image inputs, further bridging the gap between visual and textual information [13].

VI. RGAN NETWORK ARCHITECTURE

Drawing on the concept of residual connections [10], we have devised the following architecture. We use the following notation. The generator network is represented tt: $R^z * R^T \rightarrow R^I$ And the discriminator network is referenced $R^I * R^T \rightarrow \{0, 1\}\}$, where T is the size of the text description embedding, I is the dimension of the image, and Z desides the dimension of the noise input. The generator takes random noise and a text embedding as input and produces an output image. In contrast, the discriminator takes an image and a text embedding as input and classifies it as either 0 (generated image) or 1 (real image, not generated).

A. Generator
First, drawn sample from the noise prior $z \in R^z$ ~ (0,1) and encode the text query t using RNN and these are given as input to the generator. Following steps are taken by the generator:

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207044 |

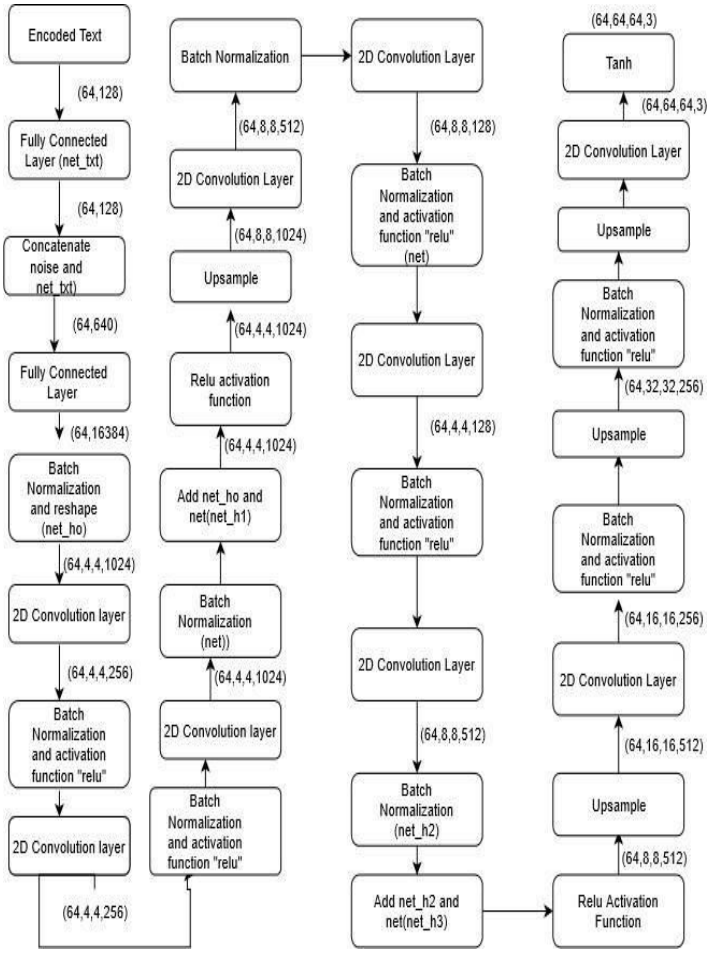

Fig. 2. RGAN Generator Architecture

B. Discriminator

The discriminator takes in an image and a written query and it outputs whether the image is generated or not. The following are the steps taken by discriminator:

Input: The discriminator receives a visual and a text query as input.
Feature Extraction: It extracts relevant features from both the depict and the text query.
Integration: These features are integrated or concatenated to form a joint representation.
Classification: Based on this joint representation, the discriminator outputs a binary decision, indicating whether the image is classified as generated (0) or real (1).

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207044 |

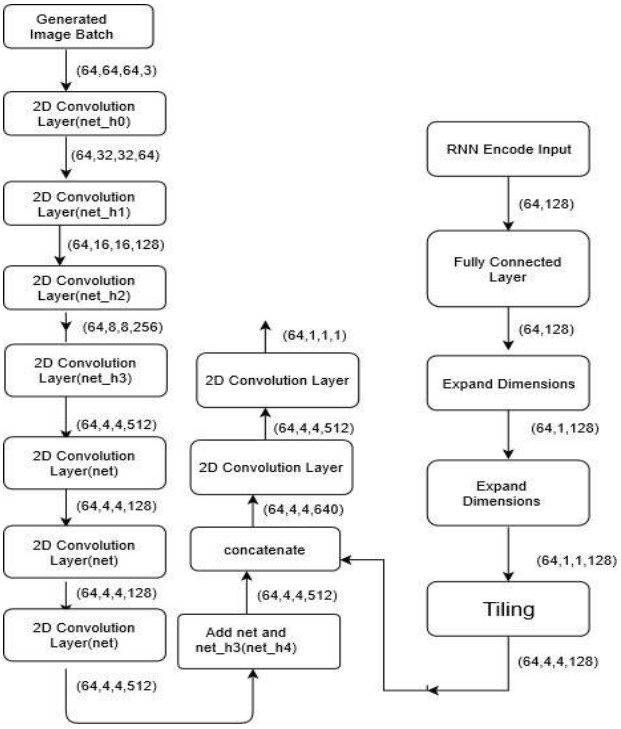

Fig. 3. RGAN Generator Architecture

VII. TRAINING ALGORITHM

To train both the generator and discriminator effectively in a GAN setup, it's crucial to balance their learning so that neither becomes too dominant. If the discriminator learns excessively, it might provide gradients that are insufficient for the generator to learn effectively. During training, the generator and discriminator operate as a unified unit cantered on the (text, image) pair. Initially, the generator produces noisy images, challenging the discriminator to distinguish between real images and these initial outputs. As training progresses, the generator refines its images to better match the visual information conditioned by the text, while the discriminator improves its ability to evaluate whether generated images meet these conditions. The discriminator is trained in a supervised manner, focusing on differentiating between:(correct text, correct image), (correct text, wrong image), and (wrong text, correct image) pairs.

RGAN Training algorithm

Input: correct image i , correct text t , incorrect image $\hat{i}$,incorrect text $\hat{t}$

For i=1 to epochs do:

For j=1 to num_batches do:

$T = \varphi(t)$ {encoding correct text using rnn}

$\hat{T} = \varphi(\hat{t})$ {encoding incorrect text}

$I = \omega(i)$ {encoding correct image}

$\hat{I} = \omega(\hat{i})$ {encoding incorrect image}

$$R_{loss} = max(0, \alpha - cos_{similarity}(I, T) + cos_{similarity}(I, \hat{T})) + max(0, \alpha - cos_{similarity}(I, T) + cos_{similarity}(\hat{I}, T))$$

$F_G = G(Z, T)$ {G takes input noise(Z) and text(T) and gives image F}

$F_{DW} = D(F_G, T)$ {D's input is Wrong(Generated) image,Correct text pair }

$F_{DC} = D(I, T)$ { D's input is Correct image,Correct text pair }

$F_{DF} = D(I, \hat{T})$ {D's input is Correct image,wrong text pair }

$G_{loss} = \gamma(F_{DW}.logits)$ {calculating generator loss}

$D_{loss} = \gamma(F_{DC}.logits) + (\gamma(F_{DF}.logits) + \gamma(F_{DW}.logits))/2$

$ADAM(G, G_{loss})$ {updating G using the loss calculated}

$ADAM(D, D_{loss})$ {updating D using the loss calculated}

$ADAM(CNN\ RNN, R_{loss})$ {updating CNN,RNN using the loss calculated}

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207044 |

The emphasis is placed on reducing the discrepancy associated with (correct text, correct image) pairs, while less weight is assigned to errors from the other pairs [1]. Unlike a typical classification network, one class's distribution (generated images) evolves over time as the generator improves. For the generator, gradients from the discriminator's output are back-propagated with respect to the generated image. These gradients are then disseminated through the generator's weights to teach it how to produce more realistic images that better fool the discriminator.

The GAN training algorithm can be condensed as follows:

1. Encode correct and incorrect images using CNN and correct and incorrect texts using RNN.
2. Calculate the RNN loss based on the similarity score between the correct text and image pair, aligning with the visual information described by the text.
3. Have the generator produce an image, which is then fed to the discriminator, representing the (correct text, wrong image) pair.
4. Feed pairs of (correct text, correct image) and (incorrect text, correct image) to the discriminator.
5. Calculate losses for both the generator and discriminator based on their respective tasks.
6. Use the Adam optimizer to modify the weights of both networks according to their losses.

This iterative process guaranties that both the generator and discriminator improve together, leading to the generation of more realistic images aligned with the given text descriptions.

VIII. EXPERIMENTS AND RESULTS

Our experiment was conducted using the Oxford 102 Flowers dataset, comprising 7,130 training images. With a batch size of 64, each epoch consisted of 115 iterations. Initially, we educated our model for 800 iterations, focusing on training the RNN and CNN for the first 80 iterations only. Subsequently, we trained both the discriminator and generator for the remaining epochs. To manage computational complexity and training time, we used 5 captions per image. This constraint allowed us to maintain acceptable image quality, although reducing the number of captions further led to a noticeable decline in image quality. We utilised 5 captions to be a balance where training time remained manageable without significant loss in image quality, though increasing to 10 captions could potentially yield better results.

For efficient model convergence, we initialized weights using the glorot initializer [3] and implemented learning rate decay, reducing the learning rate every 100 iterations. This approach helps the model to navigate closer to the minima in the loss landscape, adjusting step sizes as it approaches convergence. The dataset images are sized at 64x64 pixels with RGB channels. We set the beta1 parameter of the Adam optimizer [2] to 0.5 for momentum control. The perturbation vector dimension was 1x512, drawn from a normal distribution. Text features were represented using 128 dimensions, with word embedding's set to 256 dimensions and a vocabulary size of 8000.

During testing, we evaluated our model's performance against specific text queries extracted from image captions:
Query 1: The flower shown has yellow anther blue pistil and yellow petals.
Query 2: These flower exhibits petals that are white, and has dark lines
Query 3: The petals on this flower are pink with a dark centre.
Query 4: This flower has a lot of small round blue petals.
Query 5: This flower is red colour petals.
Query 6: The flower has dark black petals and the centre of it is brown and has black pistil.
Query 7: The flower displayed has green petals with dark centre green anther green pistil.
Query 8: These white flowers have petals that initiate as white in colour and culminate in a white towards the tips.

Evaluation of generative models lacks a standardized metric, relying instead on human judgment based on the quality of synthesized images. Below, we present our results for the aforementioned queries, showcasing the images generated by our model.

IX. CONCLUSIONS

In this study, we introduced a text-to-image architecture that incorporates skip connections and employs techniques like learning rate decay to enhance training stability and accelerate convergence. This approach has enabled the generation of images of reasonable quality that align with detailed text descriptions of visual content. Our findings demonstrate that a Generative Adversarial Network (GAN) conditioned on a text description can produce multiple images that accurately match the provided description. This capability naturally arises from the inherent multi-modality of GANs, where different plausible outputs can align with the same input description. Thus, our

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207044 |

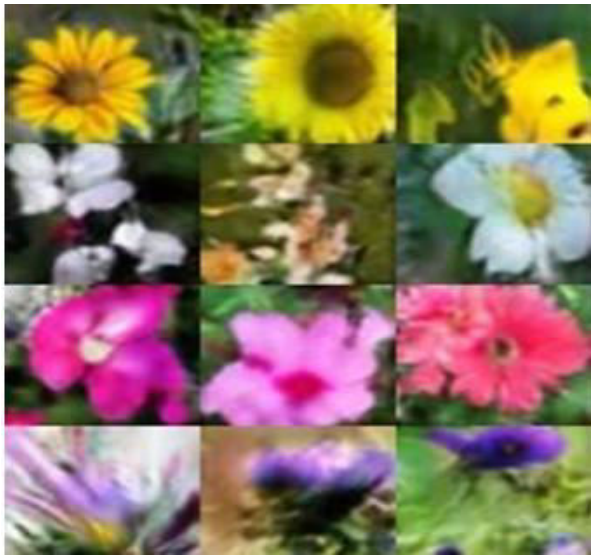

Fig 4. Results

work showcases the effectiveness of leveraging textual descriptions to guide the generation of diverse yet contextually appropriate images through the use of advanced architectural features and training methodologies.

X. FUTURE SCOPE

GANs represent a significant advancement in generating outputs that closely resemble real-world scenarios. However, there are several areas where GANs can be further developed and enhanced. These include ensuring the diversity of generated outputs, establishing robust methods for evaluating and rating these outputs, stabilizing the learning process, and devising strategies to achieve faster convergence. Currently, GANs exhibit variability in their outputs, producing multiple plausible interpretations for a given input. To improve controllability and consistency, future research needs to focus on refining techniques that enable more predictable and controlled generation processes.

CONCLUSION

Our experimental results show that while our model generates reasonably accurate images based on text descriptions, there are limitations due to using only five captions per image to manage computational and training time constraints. In future work, we plan to enlarge the model to generate higher resolution images. This scaling will involve incorporating more diverse types of textual descriptions, with regards to both background and foreground details. These enhancements aim to elevate the output quality of generated images, resulting in sharper, more detailed, and realistic outputs. By addressing these aspects, we aim to advance the capabilities of GANs in producing results that are not only visually appealing but also contextually rich and faithful to the input descriptions.

REFERENCES

1. Reed, S. E., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., & Lee, H. (2016). "Generative Adversarial Text to Image Synthesis." In Proceedings of the 33rd International Conference on Machine Learning (ICML) (pp. 1060-1069). This paper presents the foundational approach to text-to-image synthesis using GANs.
2. Zhang, H., Xu, T., Li, H., Zhang, S., Huang, X., Wang, X., & Metaxas, D. N. (2017). "StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks." In Proceedings of the IEEE International Conference on Computer Vision (ICCV) (pp. 5907-5915). This study introduces StackGAN, a two-stage GAN for synthesizing high-quality images from text descriptions.
3. 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 (CVPR) (pp. 1316-1324). AttnGAN improves the text-to-image synthesis process by focusing on fine-grained details.
4. Li, J., Zhang, X., Liu, J., & Li, H. (2019). "StoryGAN: A Sequential Conditional GAN for Story Visualization." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 6329-6338). This paper explores the generation of sequential images from text, advancing the concept of story visualization.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207044 |

5. Miyato, T., Kataoka, T., Koyama, M., & Yoshida, Y. (2018). "Spectral Normalization for Generative Adversarial Networks." arXiv preprint arXiv:1802.05957. This paper discusses a technique to stabilize the training of GANs, which is crucial for generating high-quality images from text.
6. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). "Generative Adversarial Nets." In Advances in Neural Information Processing Systems (NeurIPS) (pp. 2672-2680). This is the seminal paper that introduced GANs, laying the groundwork for text-to-image synthesis models.
7. Zhang, H., Goodfellow, I., Metaxas, D., & Odena, A. (2019). "Self-Attention Generative Adversarial Networks." In Proceedings of the 36th International Conference on Machine Learning (ICML) (pp. 7354-7363). This study incorporates self-attention mechanisms into GANs to improve image generation quality.
8. Chen, Y., Li, Y., & Modarres, M. (2020). "Text2Image: Deep Learning Techniques for Text-Driven Image Generation." IEEE Access, 8, 100287-100298. This paper reviews various deep learning techniques for text-driven image generation, providing a comprehensive overview.
9. Han, Z., Wang, J., Shao, L., & Zheng, N. (2020). "GAN-based Image Synthesis for Deep Learning in Computer Vision." ACM Computing Surveys, 53(2), 1-30. This survey paper covers GAN-based image synthesis applications and their impact on deep learning tasks in computer vision.
10. Dai, B., Fidler, S., Urtasun, R., & Lin, D. (2017). "Towards Diverse and Natural Image Descriptions via a Conditional GAN." In Proceedings of the IEEE International Conference on Computer Vision (ICCV) (pp. 2970-2979). This research explores generating diverse and natural image descriptions, relevant for text-to-image synthesis

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details