iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** IV **Month of publication:** April 2023

DOI: https://doi.org/10.22214/ijraset.2023.50584

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue IV Apr 2023- Available at www.ijraset.com

Text to Image Synthesis using Generative Adversarial Networks

Anushree Dandekar[1], Rohini Malladi[2], Payal Gore[3], Dr. Prof. Vipul Dalal[4]
Department of Information Technology, Department of Vidyalankar Institute of Technology, Mumbai, India

Abstract: ***Image generation has been a significant field of research in computer vision and machine learning for several years. It involves generating new images that resemble real-world images based on a given input or set of inputs. This process has a wide range of applications, including video games, computer graphics, and image editing. With the advancements in deep learning, the development of generative models has revolutionized the field of image generation. Generative models such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) have demonstrated remarkable success in generating high-quality images from input data. The focus of this paper is to propose a new technique for generating high-quality images from text descriptions using Stack Generative Adversarial Networks (StackGAN). Through a sketch-refinement process, the problem is also divided into smaller manageable problems. The proposed StackGAN model comprises two stages, Stage-I and Stage-II. Stage-I GAN generates low-resolution images by sketching the primitive shape and colors of the object based on the provided textual description. Stage-II GAN generates high-resolution photo-realistic images with refined details by taking the Stage-I results and textual descriptions as inputs, along with detecting defects and adding details.***

I. INTRODUCTION

Generating images from text has become a popular trend in recent years due to the increasing demand for creative and personalized visual content. This technology enables the creation of photo-realistic images based on textual descriptions, providing endless possibilities for various applications such as virtual and augmented reality, video games, and social media. With the advancement of deep learning techniques and the availability of large datasets, researchers have developed various methods to generate high-quality images that accurately reflect the intended meaning of the textual input. This trend is expected to continue growing as the technology advances and becomes more accessible to a wider audience, revolutionizing the way we create and consume visual content. Generation of images can be performed using deep learning technologies like Generative Adversarial Networks. Generative Adversarial Networks (GANs) [8] are advanced neural networks that involve multiple networks competing with each other to produce highly accurate and nearly indistinguishable rendered images. These networks function using game theory, where a generator network creates images and a discriminator network classifies the image as authentic or fake. Through training, the generator learns to produce better and more realistic images, eventually leading to convergence where authentic images are reliably generated. For the process of synthesizing high-quality images from text descriptions, we propose the use of Stack Generative Adversarial Networks (StackGAN) [9] which decomposes the process into manageable sub-processes. Our Stage-I GAN generates low-resolution images based on text descriptions, which are then refined and improved by our Stage-II GAN to produce photo-realistic high-resolution images. The resulting images are of high quality and can be used in a variety of practical applications.

In Section II, Literature Review of the paper is discussed wherein, technical research papers and some existing systems were studied. In Section III, Proposed Approach is elaborated. In Section IV, the dataset and the implementation part of the system are discussed in detail. In Section V, the results are presented and discussed. In Section VI, future scope of the project citing various ways of project application is mentioned. In Section VII, conclusions are stated. In Section VIII, references are quoted.

II. LITERATURE REVIEW

Deep learning techniques have made remarkable progress in generating images from text.

1) *Variational Autoencoders (VAE)* : Variational Autoencoders (VAEs) are a type of neural network that can learn to represent complex data like images and audio in a compact way. They use probabilistic techniques to encode the input data into a lower-dimensional space, called a latent space. This allows them to generate new data that is similar to the original input data. [1]
2) *AlignDRAW* : AlignDRAW is an image generation model that aligns text and image features to produce high-quality images from textual descriptions. The model uses an attention mechanism to align the text and image features and generates the image in a step-by-step fashion. AlignDRAW was introduced by Mansimov et al. in 2016 and has shown promising results in generating realistic images from textual descriptions. [2]

©IJRASET: All Rights are Reserved

3) *Conditional PixelCNN* : Conditional PixelCNN is an autoregressive model that generates images conditioned on a given input, such as text descriptions or object location constraints. It models the conditional distribution of the pixel space using a convolutional neural network. [3, 11]
4) *Laplacian pyramid framework* : Laplacian pyramid framework is a multi-scale image decomposition technique that involves dividing an image into several levels, with each level representing a specific scale. This framework is used in image processing and computer vision applications to reconstruct high-resolution images from low-resolution inputs. [4] Various methods [15, 16] have been suggested to improve the stability of the training procedure and produce impressive outcomes. A GAN model based on energy has been suggested to enhance the stability of the training process. [17]
5) [7] This paper proposes a deep neural network that predicts image pixels along two spatial dimensions, modeling the distribution of natural images. This method achieves better log-likelihood scores on ImageNet and generates coherent, varied images. It aims to bridge the gap between advances in text and image modeling through a novel deep architecture and GAN formulation. By utilizing powerful recurrent neural network architectures and deep convolutional GANs, the model can generate realistic images of birds and flowers from text descriptions.
6) [5] This paper aims to present an overview of image synthesis with GAN, highlighting the strengths and weaknesses of current methods. The main approaches are classified into direct, hierarchical, and iterative methods, with a discussion on text-to-image synthesis and image-to-image translation. It provides guidance for those applying GAN to their problems and aims to advance research in GAN for artificial intelligence.
7) [6] The paper proposes Stack Generative Adversarial Networks (StackGAN) to generate photo-realistic images of 256x256 resolution based on text descriptions. The approach decomposes the problem into two sub-problems and introduces Conditioning Augmentation to improve diversity and stability. The proposed method outperforms state-of-the-art approaches on benchmark datasets. StackGAN, has an advantage over the aforementioned models as it utilizes a multi-stage GAN architecture that can generate high-resolution images with fine details while maintaining textual consistency. It also effectively captures both low-level and high-level features of the image and produces realistic images that are visually and semantically meaningful.

III. PROPOSED APPROACH

A. *Generative Adversarial Networks (GAN)*

GANs utilize two neural network models, the generator and discriminator, to produce data through adversarial learning [10, 13]. The generator takes in random noise (z) and creates data, while the discriminator differentiates between real and synthetic data generated by the generator. The generator's goal is to produce data that can fool the discriminator, which is trained to recognize the source data.

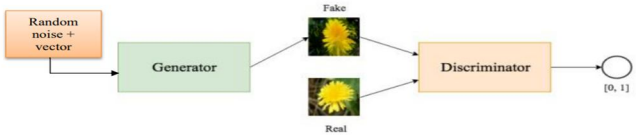


Figure 1: GAN

B. *Stack Generative Adversarial Networks (StackGAN)*

Model Architecture of StackGAN consists of mainly the following components:

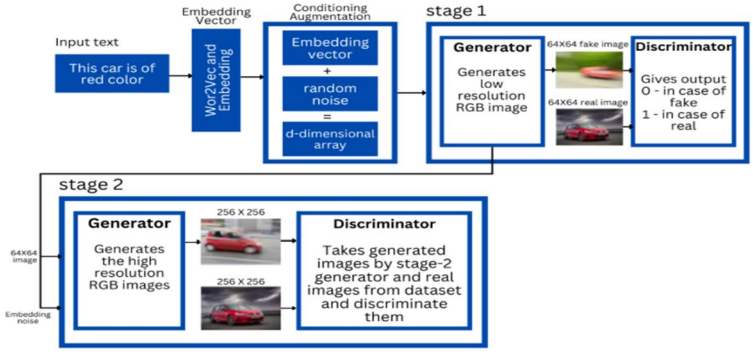


Figure 2: StackGAN architecture

1) *Embedding*: Converts the input variable length text into a fixed length vector. We will be using a pre-trained character level embedding. In StackGAN, embeddings [13] are used to convert text descriptions into numerical vectors that can be used as input for the generator network. The embeddings are learned through a separate neural network trained on a large text corpus. These embeddings capture semantic information about the text, allowing the generator network to produce images that align with the text description. The use of embeddings improves the quality and diversity of the generated images by better capturing the relationship between the text and the image.
2) *Conditioning Augmentation (CA)* : Conditioning Augmentation (CA) [13] is a technique used in generative models to improve the diversity and quality of generated samples. It involves adding random noise vectors to the conditioning variables during training, which encourages the model to learn a more robust representation. In StackGAN, CA is used to generate multiple variations of the same text description, which allows for a greater range of images to be generated. This helps to address the issue of mode collapse, where a model generates only a limited set of similar outputs.
3) *Stage I Generator* : The stage I generator of StackGAN is a conditional GAN that generates a low-resolution image from a given text description. It takes a random noise vector and the text embedding as input, and outputs a 64x64 image. The generator is trained to produce images that match the given text description while maintaining basic color and shape constraints. The generated image is then fed to the stage II generator for refinement and adding more details. Overall, the stage I generator is responsible for generating the initial sketch of the image based on the text input.
4) *Stage I Discriminator* : The Stage-I discriminator of StackGAN takes in the generated low-resolution image and the text embedding as inputs and then applies a series of convolutional layers to learn the features of the image. The discriminator is trained to distinguish between the generated image and the real image. It encourages the generator to produce images that are similar to the real images in terms of their content and style. The adversarial loss is calculated based on the output of the discriminator and used to update the parameters of the generator. The Stage I discriminator plays an important role in ensuring that the generated images follow the basic color and shape constraints of the given text description.
5) *Residual Blocks* : In StackGAN, residual blocks are used in both Stage I and Stage II generators to improve the quality of generated images. Residual blocks allow the network to learn residual features that capture the difference between the input and output. This helps to avoid the problem of vanishing gradients and enables deeper networks to be trained effectively. The residual blocks in StackGAN are designed to preserve the spatial resolution of the image features while increasing their channel depth, thus allowing for more complex and detailed feature representations. Overall, the use of residual blocks in StackGAN helps to improve the quality and resolution of generated images.
6) *Stage II Generator* : The Stage II generator of StackGAN takes the low-resolution image generated by the Stage I generator and enhances it with more detailed and realistic features. It consists of two parts: the first part generates a rough estimate of the high-resolution image, and the second part refines the generated image with more details using residual blocks. The Stage II generator is conditioned on the same text description as the Stage I generator, and it also receives an additional conditioning vector generated by the Conditioning Augmentation technique, which adds more diversity to the generated images. The final output of the Stage II generator is a high-resolution image with photo-realistic details and high diversity.
7) *Stage II Discriminator* : The Stage II Discriminator in StackGAN is designed to discriminate between real and generated high-resolution images. It takes the high-resolution images generated by the Stage II Generator as input and produces a score indicating how closely the generated image matches the real images from the training dataset. The architecture of the Stage II Discriminator is similar to that of the Stage I Discriminator, but it is designed to handle higher resolution images with more detailed features.

IV. EXPERIMENTAL SETUP

This section describes the Dataset and the Implementation.

A. Dataset

To train our model, we have selected the Caltech-UCSD Birds [12] dataset that consists of 11,788 images of 200 different bird species. For every image in the CUB dataset, Caltech has provided 10 corresponding descriptions. This dataset will be used to prepare our model for the task of generating photo-realistic images based on textual descriptions. By utilizing this dataset, our model will be able to learn and recognize the unique characteristics of each bird species, and generate images that are consistent with the provided textual descriptions. The large size and diversity of the CUB dataset will also help ensure that our model is able to generate images with high resolution and quality, and that it can handle a wide range of different bird species and descriptions. For example, the image in Figure 3 shows a cactus wren, and the text describing this image is presented below.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue IV Apr 2023- Available at www.ijraset.com



Figure 3: Example image from Caltech-UCSD Birds.

"a medium bird with a black body, white back and a peach crown."

B. Implementation

The implementation of the project was done using the following resources –

1) Jupyter notebook
2) NVIDIA Cuda
3) Tensorflow

Some of the main functions implemented are –

1) *Conditioning Augmentation* - Conditioning augmentation is a method in machine learning that enhances model performance by adding supplementary information to input data during training.
2) *UpSampling* - An upsampling block in a Generative Adversarial Network (GAN) is a component that increases the resolution or size of generated images by enlarging low-resolution feature maps to generate higher-resolution images.
3) *Conv Block* - Convolution 2D in a Generative Adversarial Network (GAN) refers to a type of operation that applies a filter or kernel to input image data to extract relevant features and generate output feature maps, which are then used to generate images.
4) *Building embedding compressor* – An embedding compressor model in a Generative Adversarial Network (GAN) is a component that is used to reduce the dimensionality of input embeddings, typically used for generating images, while preserving their semantic information, resulting in a compressed representation.
5) A*dversarial loss function* – Adversarial loss in a Generative Adversarial Network (GAN) is a measure of the discrepancy between the generated and real data distributions, calculated using a discriminator network, which is trained to distinguish between the two.
6) *Residual Block* – A residual block function in a Generative Adversarial Network (GAN) is a type of building block that allows for the flow of both original input and residual information through multiple layers, aiding in the optimization and learning of complex mappings while mitigating the vanishing gradient problem.
7) *Downsampling* – A downsampling function in a Generative Adversarial Network (GAN) is a process that reduces the spatial dimensions or resolution of input data, typically through operations like pooling or strided convolution, to extract lower-dimensional representations or feature maps for generating downsampled images or data.
8) *ReLU* – Rectified Linear Unit is a common activation function in neural networks that increases the speed of training and is used to introduce non-linearity by allowing positive values to pass through while zeroing out negative values.
9) *Batch normalization* – is a technique that normalizes the input data to prevent overfitting and improve training speed. It also improves the stability. [13]
10) *Tanh function* – is an activation function that maps input values to a range of -1 to 1. It is used to produce a bounded and continuous output that can represent diverse data distributions.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

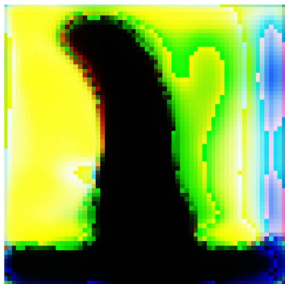
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue IV Apr 2023- Available at www.ijraset.com

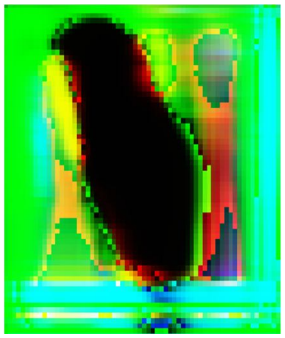
11) LeakyReLU – is an activation function allows some negative input values to pass through to avoid the "dying ReLU" problem.
After stage I network is trained using the mentioned data set and the images are generated, they are used as input for the training of stage II. The first stage of the generator creates a low-resolution image by drawing the rough shape and colors from the text and painting the background with noise, while the second stage adds details and corrections to produce a more realistic high-resolution image.

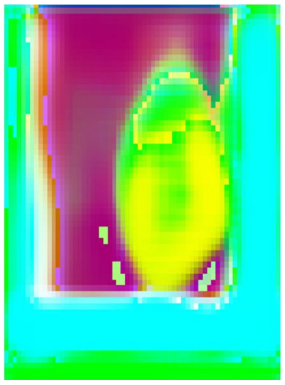
V. RESULTS

The functions and methodology mentioned above were executed and images were generated by training the model. The project is focused on generating photo-realistic images from textual description using Stack Generative Adversarial Network.
As mentioned in the proposed approach, the training of the model was done in two stages. Stage I took 12 hours for 36 epochs. Meanwhile stage II, requiring high performance computation power than stage I, took 18 hours for 8 epochs.
In both the stages, 10 images were saved for every third epoch. Accordingly, at the end of 36 epochs in stage I, we had 120 images saved.

Following are some of the images generated in stage I –


Figure 4: Epoch 21, Image 5


Figure 5: Epoch 24, Image 8


Figure 6: Epoch 30, Image 9

©IJRASET: All Rights are Reserved

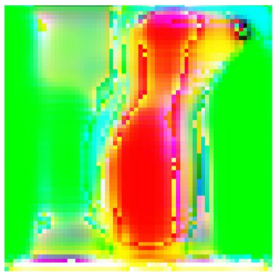


Figure 7: Epoch 36, Image 8

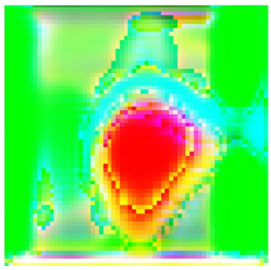


Figure 8: Epoch 36, Image 10

Though the images after stage I looked a little promising, stage II results after just 8 epochs were not up to the required standard. It needed a more powerful machine with a processing power much greater than the machine we had. Due to hardware constraints, we were able to run less number of epochs, however, we believe that greater number of epochs will assure higher quality images.

Following are some of the images generated in stage II –



Figure 9: Epoch 3, Image 1

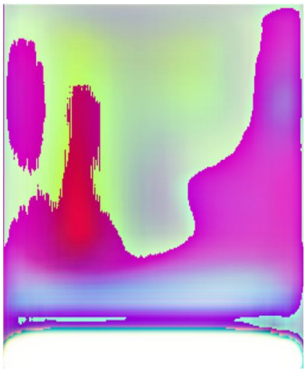


Figure 10: Epoch 3, Image 8

VI. FUTURE SCOPE

Generating images from text has a variety of applications and innumerable ways in which it can be useful in multiple ways. Some of the ways are:

1) *Editing*: The use of image generation from text in the field of photo editing allows for automated image editing based on natural language descriptions, which can save time and resources compared to traditional manual editing. With this technology, photo editors can quickly generate new image variations based on client preferences or design requirements. Additionally, it can be used to generate realistic-looking images of products or locations that do not yet exist, which can be helpful for marketing and advertising purposes.
2) *Interior Designing:* Image generation from text can be beneficial in interior designing as it can help designers create realistic and personalized visualizations of their designs before actually implementing them. This can save time and resources as it enables them to experiment with different color schemes, furniture arrangements, and lighting options. It can also help clients visualize their space and make informed decisions about the design. Moreover, the generated images can be used in marketing materials and presentations to showcase the design to potential clients.
3) *AR VR:* The use of image generation from text in augmented reality and virtual reality allows for the creation of realistic and customizable virtual environments and objects, improving the immersive experience of users. By generating images from textual descriptions, AR and VR applications can offer a greater degree of flexibility and personalization, as well as reduce the need for expensive and time-consuming manual design processes. This technology can also be applied to generate virtual representations of real-world objects, such as furniture or buildings, which can be useful for architectural visualization or product demonstrations.
4) *Search Engines:* The use of "image generation from text" in search engines can help improve the search experience by generating relevant images from textual queries. This can enhance the accuracy and completeness of search results, making it easier for users to find the information they are looking for. Additionally, image generation from text can aid in visual search, where users can input a description of an image they are looking for and have the search engine generate relevant images based on that description. This technology can also have applications in image retrieval systems and content-based image retrieval.

VII. CONCLUSION

In conclusion, the proposed method of using Stack Generative Adversarial Networks (StackGAN) with Conditioning Augmentation shows promising results in synthesizing photo-realistic images from text. The use of Stage-I and Stage-II GANs allows for the creation of higher resolution images with more photo-realistic details, surpassing other text-to-image generative models. This technique has significant potential for use in various fields such as interior designing, virtual reality, and assistive communication tools. As technology advances and more complex datasets become available, the application of StackGAN with Conditioning Augmentation can lead to even more impressive results in generating realistic images from text descriptions.

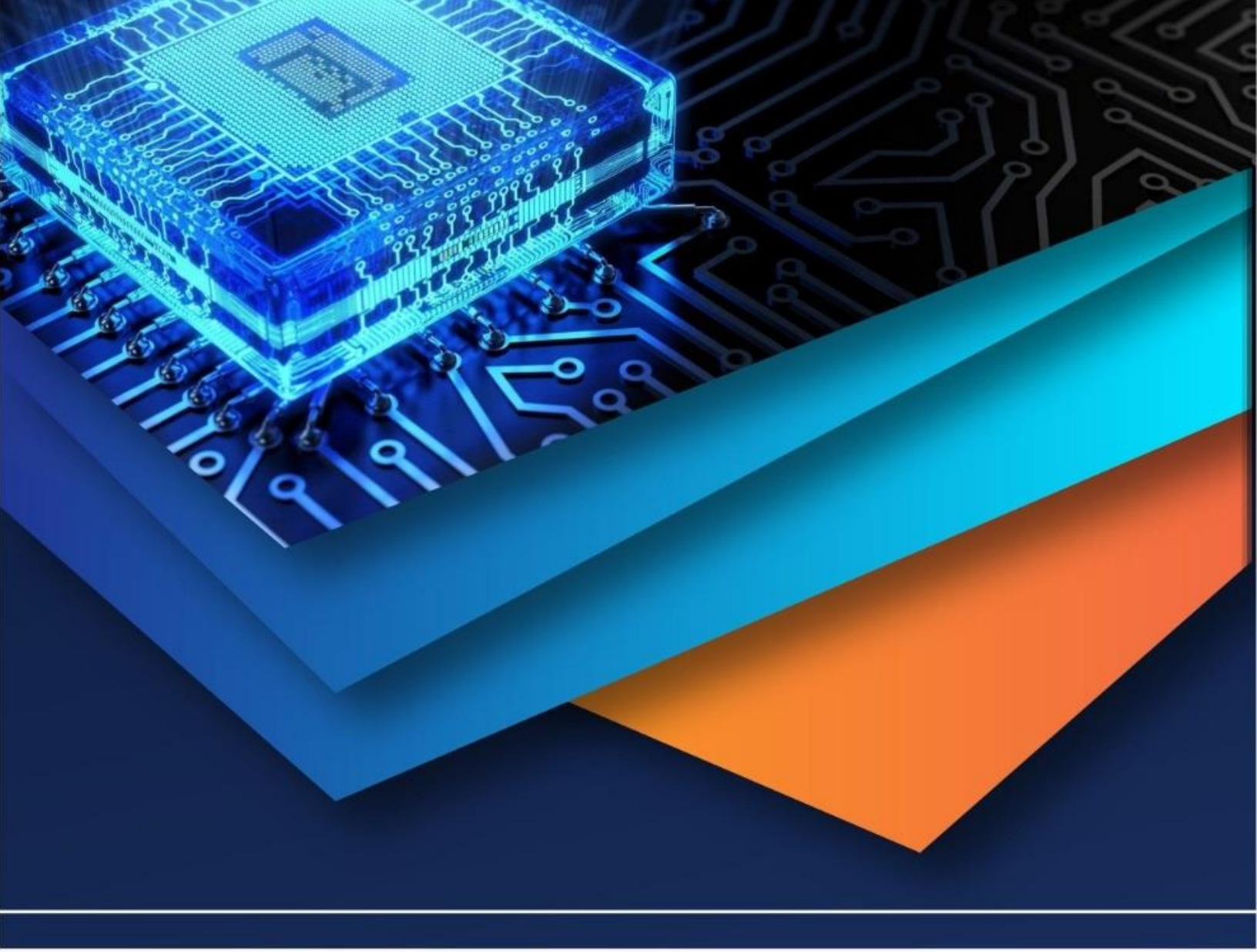
REFERENCES

[1] Doersch, “Tutorial on variational autoencoders”. arXiv preprint arXiv:1606.05908 [stat.ML], pp. 4-7, Jan 2021
[2] E. Mansimov, E. Parisotto, L. J. Ba, and R. Salakhutdinov, "Generating images from captions with attention," in International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, 2016, pp.5-7.
[3] S. Reed, A. van den Oord, N. Kalchbrenner, V. Bapst, M. Botvinick, N. de Freitas. “Generating interpretable images with controllable structure”. in International Conference on Learning Representations (ICLR), Toulon, France, 2017, pp.3-6.
[4] E. L. Denton, S. Chintala, A. Szlam, and R. Fergus. “Deep generative image models using a laplacian pyramid of adversarial networks”. in Conference on Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 2015, pp.2-4
[5] H. Huang, P. S. Yu, and C. Wang, "An Introduction to Image Synthesis with Generative Adversarial Nets," IEEE Signal Processing Magazine, vol. 37, no. 3, pp.6-10, May 2020.
[6] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D.N. Metaxas, "StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks," in 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp.1-9.
[7] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative Adversarial Text-to-Image Synthesis," in Proceedings of the 33rd International Conference on Machine Learning (ICML), New York, NY, USA, 2016, pp.3-8.
[8] C. Bodnar, "Text to Image Synthesis Using Generative Adversarial Networks," arXiv:1605.05396, pp.33-55, May 2016.
[9] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie, "Stacked generative adversarial networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp.2-7.
[10] S. Frolov, T. Hinz, F. Raue, J. Hees, and A. Dengel, "Adversarial Text-to-Image Synthesis: A Review," arXiv:1910.13145, Oct. 2019, pp.3-16.
[11] A. van den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu, "Conditional Image Generation with PixelCNN Decoders," in Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS), Barcelona, Spain, Dec 2016, pp.3-6.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

[12] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, Technical Report CNS-TR-2011-001, 2011.
[13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in Proceedings of the International Conference on Machine Learning (ICML), 2015, pp.2-7.
[14] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in Proceedings of the International Conference on Machine Learning (ICML), 2016, pp.3-7.
[15] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in Proceedings of the International Conference on Learning Representations (ICLR), 2016, pp.2-8.
[16] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in Proceedings of the Neural Information Processing Systems (NIPS), 2016, pp.2-6.
[17] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based generative adversarial network," in Proceedings of the International Conference on Learning Representations (ICLR), 2017, pp.2-12.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)