Text to Photo-Realisitic Image Construction

Venkata Surya Sai Ram Nistala  
*University of Central Missouri*Lee’summit, U.S  
[vxn71470@ucmo.edu](mailto:esha.latief@s.amity.edu)

Neeraj kumar Kajuluri  
*University of Central Missouri*Lee’summit, U.S  
[nxk20910@ucmo.edu](mailto:esha.latief@s.amity.edu) Vijitha Gajavelli  
*University of Central Missouri*Lee’summit, U.S  
[vxg97070@ucmo.edu](mailto:pkumar4@amity.edu)

Krishna Sai Srujan Teja Basetty  
*University of Central Missouri*Lee’summit, U.S  
[k](mailto:sunnyanand.42@gmail.com)xb52990@ucmo.edu

***Abstract*—*One of the computer vision trends with the fastest growth and highest computational demands is image synthesis. Although Generative Adversarial Networks (GANs) have demonstrated amazing performance in a variety of tasks, they still confront hurdles in producing high-quality photographs. In this study, we present Stacked Generative Adversarial Networks (Stack Gans) for creating high-resolution photorealistic photographs. First, we present StackGAN-v1, a two-stage generative adversarial network architecture for text-to-image synthesis.***

***The goal of text-to-photorealistic image construction is to create a photo-realistic image from a given natural language description. The creation of images from text is one of the use cases for Generative Adversarial Networks (GANs), which have many industrial applications. Under different conditions and circumstances, the generation of images using GANs was very successful. Natural language's expressiveness brought text-to-image construction research to public attention. But creating high-quality, photorealistic images that also meet all the requirements stated in the text Descriptions is extremely difficult and challenging. We are evaluating and contrasting cutting-edge algorithms Stack GAN, Stack GAN++, DALL-E, and DALL-E 2 as part of text to photo-realistic image construction.***

***Creating high-quality images from text descriptions is a difficult computer vision problem that has many different solutions. Applications that are useful. The samples created by current text-to-image approaches can roughly capture the meaning of the provided descriptions, but crucial details and vivid object parts are absent. We divide the difficult problem into more manageable subproblems using a sketch-refinement method. Stage-I GAN creates Stage-I low-resolution images by sketching the basic shape and colors of the object based on the provided text description. The Stage-II GAN produces high-resolution images with lifelike features using the Stage-I results and text descriptions as inputs. It is able to correct issues with Stage-I results and add enticing details through the refinement process. In order to increase the variety of the synthesized images and stabilize the conditional-GAN training, we present a novel conditioning augmentation technique that encourages smoothness in the latent conditioning manifold. The proposed method significantly enhances the ability to generate photorealistic images based on text descriptions, according to extensive tests and comparisons with leading technologies on benchmark datasets.***

Keywords—GAN, CGAN, Stack GAN, Stack GAN++, Image Synthesis, CLIP.

# **Introduction**

With the advent of Stacked Generative Adversarial Networks (Stack GAN), a deep learning architecture, which uses a GAN conditioned on text embeddings, text-to-image synthesis has long been a popular area of research [1-3]. Han Zhang and colleagues first proposed it in 2017 [4]. Both Stack GAN and Stack GAN++ (also known as Stack GAN v2) scale up the image resolution and produce high-quality images with fine-grained details by using multi-scale GANs [5].

The Stack GAN architecture employs two GANs that are stacked one on top of the other to create a network that can produce high-resolution and high-quality images [6]. Stage I and Stage II are its two stages. The Stage-I network creates low-resolution images with basic colors and rough sketches using text embedding with constraints, while the Stage-II network uses the images created by the Stage-I network and scales them up to create high-resolution images based on the provided text embeddings [7]. The second network fixes flaw and adds minute details to produce a higher-resolution image that is more realistic. In a Stack GAN, Stage-I basically generates basic shapes while Stage-II fixes’ errors in the image that the Stage-I network produced. To make the image appear more photo-realistic, Stage-II also adds more detail [8]. In both stages, generator networks called Conditional Generative Adversarial Networks (CGANs) are employed. The first stage's Conditional GAN is dependent on the text descriptions, whereas the second stage's Conditional GAN is dependent on both the text descriptions and the images produced by the first GAN [9].

For both conditional and unconditional generative tasks, Stack GAN++ has a sophisticated multi-stage generative adversarial network architecture [10]. A cascaded refinement network, which adds more details and enhances the visual quality of the generated image, extends the original Stack GAN architecture. In contrast to StackGAN, which only has two generators, Stack GAN++ has multiple generators that are organized like a tree and share many of the parameters [11]. Each generator is then treated as a branch of the tree. The text embedding that forms the network's input can be thought of as the tree's base, and each branch—or generator—produces images at a different scale, with the deepest generator producing high-resolution, photorealistic images that are used as outputs while intermediate generators produce images at scales ranging from small to large as they get closer to the deepest generator [12-14]. The image distributions of each generator in the network are improved using feedback from training the network, and all the generators in the network are trained simultaneously. The StackGAN++ architecture uses two complementary types of image distributions that are simultaneously trained [15]. The distribution of images can be done in two ways: unconditionally or with or without text descriptions. This architecture also has a color-consistency regularization term that aids the generators in producing various image samples at various scales and colors [16].

The image distributions of each generator in the network are improved using feedback from training the network, and all the generators in the network are trained simultaneously. The StackGAN++ architecture uses two complementary types of image distributions that are simultaneously trained. The distribution of images can be done in two ways: unconditionally or with or without text descriptions. This architecture also has a color-consistency regularization term that aids the generators in producing various image samples at various scales and colors [17].

The Stacked Generative Adversarial Networks have made the following significant contributions, to sum it up. (i) Text descriptions for 256x256 resolution photorealistic images are produced by StackGAN. (ii) By using multiple generators and image distributions to create high-quality, photo-realistic images at various scales and colors, StackGAN++ enhances the original model even more.

Open AI created the deep learning models or programs DALL-E and DALL-E 2 to create photorealistic images from text descriptions.

In 2021, Open AI unveiled DALL-E, a program that creates original, photorealistic images from text descriptions. It creates pictures of abstract ideas in addition to pictures of everyday objects [18]. The GPT (Generative Pre-trained Transformer) model was first created by Open AI in 2018. A language model called GPT, which takes an incomplete sentence as input and generates the rest of the sentence, was expanded and released as GPT-2 by Open AI in 2019. The GPT-2 model is scaled up even more to create the 175 billion parameter GPT-3 model. A GPT-3 model variant with 12 billion parameters is used by DALL-E. A text generation model called the GPT-3 model produces text that resembles human writing. It was trained using a sizable text dataset. The GPT-3 model's ability to generate text is extended by DALL-E so that it can also produce images from text. A large dataset of image-text pairs encoded into 1280 tokens (1024 tokens for image representation and 256 tokens for text representation) served as the training data for the transformer language model DALL-E. In addition to creating an entire image, DALL-E can also alter a specific area of an existing image using text inputs [19].

In 2022, Open AI unveiled DALL-E 2, a replacement for DALL-E that would produce more realistic images at a 4X higher resolution. DALL-E 2 architecture uses a diffusion model based on CLIP image embeddings and has 3.5 billion parameters [20].

A model created by Open AI called CLIP (Contrastive Language-Image Pre-Training) is trained to recognize the similarities and differences between images and the text descriptions that go with them, with the goal of achieving a similarity score that is high for the images and the text descriptions that go with them and a score that is lower for the images and other text descriptions. In the architecture of DALL-E 2, CLIP is crucial. By feeding CLIP text embeddings into an earlier model, CLIP image embeddings can be created. In addition to having zero-shot capabilities and achieving cutting-edge results on vision and language tasks, CLIP embeddings are resilient to changes in image distribution. Images and videos are produced using diffusion models. Model DALL-E 2 has two stages. The given text description is first encoded to a CLIP text embedding by a text encoder. The prior model created CLIP image embedding in stage one using this CLIP text embedding as input. The diffusion model (decoder) creates the image in stage two using this CLIP image embedding as input. As a decoder, this model tried both the autoregressive and the diffusion models; however, the diffusion model proved to be more computationally effective and to produce images of high quality. The diffusion model creates an image output after being trained by receiving a CLIP image embedding as input, and it produces multiple images for each individual CLIP image embedding. By providing text description as input, the prior model is trained to produce CLIP image embedding.

# **Motivation**

Another possible use is the development of visual assistance for those with vision problems. It may be feasible to offer these people a more thorough awareness of their environment by producing visuals from written descriptions. A system that creates photo-realistic representations of a room from a written description, for example, might assist a blind person in navigating their surroundings more successfully.

Overall, the goal of text-to-photorealistic picture building is to allow for more seamless integration of natural language comprehension with computer vision. Researchers want to create new apps and technologies that will help people better comprehend and interact with the environment around them by creating algorithms that can produce realistic visuals from textual descriptions

# **Main Contribution and Objectives**

This project uses four algorithms, and everyone is involved in every step of it. We have all completed the following tasks:

* VIJITHA: Investigating everything and composing a thorough project report, researching the DALL-E2 architecture.
* SRUJAN: Complete analyses of DALL E and DALL E2, along with a brief attachment of that data and concepts.
* SAIRAM: Applying and assessing StackGAN analysis, execution, and programming, with the outcomes being added to the reports, StackGAN++ deployment and training.
* NEERAJ: The application of StackGAN++ (StackGAN-V2) analysis, execution, and programming is evaluated, and the outcomes are then added to the reports.

# **Related work**

We are examining and contrasting 4 different algorithms as part of the text to photo-realistic image construction.

1. StackGAN

2. StackGAN++ (StackGAN-V2)  
3. DALL-E

4. DALL-E 2

### **StackGAN**

The StackGAN architecture employs two GANs that are stacked one on top of the other to create a network that can produce high-resolution and high-quality images. Stage I and Stage II are its two stages. The Stage-I network creates low-resolution images with basic colors and rough sketches using text embedding with constraints, while the Stage-II network uses the images created by the Stage-I network and scales them up to create high-resolution images based on the provided text embeddings.

Diagram

Description automatically generated

***Fig. 1. The architecture of StackGAN***

The text description is first transformed into a text embedding, and then, in the Conditioning Augmentation (CA) section, the text embedding is combined with random noise. This is given to Stage-I Generator as input, and it up samples the data to create 64x64 images. Stage-I Discriminator will be given this 64x64 image and text embedding to determine whether the image is real or fake (created by Stage-I generator). The 256x256 high-resolution image produced by Stage-I generator using the 64x64 image and text embedding will be fed into Stage-II generator using some down sampling, residual, and up sampling blocks. The Stage-II Discriminator will determine whether the 256x256 image is real or fake (it was created by the Stage-II generator) by passing it along with text embedding. The image and text embedding pairs will be the input for the discriminator (Stages I and II). When given a real image and its corresponding text embedding as input, the discriminator is trained to produce an output of 1. If the input is an incorrect image or a generated image with matching text embedding, the output will be 0.

### **StackGAN++**

Diagram

Description automatically generated

***Fig. 2. The architecture of StackGAN++***

By including a cascaded refinement network, which adds more details and enhances the visual quality of the generated image, StackGAN++ extends the original StackGAN architecture. StackGAN++ has more GANs than StackGAN, which only has two, and the generators are connected in a tree-like structure. In contrast to StackGAN, where generators are trained sequentially, all the generators in this network are trained simultaneously.

Three pairs of generators and discriminators are used in the architecture above. The text embedding is created first from the text description. A random noise vector was added to this text embedding and supplied as input to the G0 generator. A 64x64 image is produced by a series of up sampling blocks in the Generator G0. This 64x64 image and text embedding were passed as input to generator G1, which produces a 128x128 image by using residual blocks and an up-sampling block. A 256x256 image is produced from this 128x128 image and text embedding when they are passed as inputs to generator G2, which is like G1. Every generator has a discriminator attached to it that produces a 1 as the output if the input is the actual image and its corresponding text embedding. If the input is an incorrect image or a generated image with matching text embedding, the output will be 0.

### **DALL-E**

A GPT-3 model variant with 12 billion parameters is used by DALL-E. A text generation model called the GPT-3 model produces text that resembles human writing. It was trained using a sizable text dataset. The GPT-3 model's ability to generate text is extended by DALL-E so that it can also produce images from text. A large dataset of image-text pairs encoded into 1280 tokens (1024 tokens for image representation and 256 tokens for text representation) served as the training data for the transformer language model DALL-E.

Diagram

Description automatically generated

***Fig. 3. The architecture of VAE***

Two stages make up the DALL-E architecture. A VAE (Variational Auto Encoder) is used in Stage 1. An autoregressive transformer is stage 2. Stage 1 of VAE training involves training it to reduce 256x256 images to 32x32 size image tokens. To train the autoregressive transformer in Stage 2, 1024 tokens (32x32) from Stage 1 and 256 encoded text tokens are combined into a total of 1280 tokens.

### **DALL-E 2**

DALL-E 2 is designed to generate more realistic images at 4X higher resolutions than DALL-E. DALL-E 2 architecture uses a diffusion model reliant on CLIP image embeddings and has 3.5 billion parameters. A model that has already been trained to recognize the similarities and differences between images and their corresponding text descriptions is called CLIP (Contrastive Language-Image Pre-Training). By feeding CLIP text embeddings into an earlier model, CLIP image embeddings can be created. Model DALL-E 2 has two stages. The given text description is first encoded to a CLIP text embedding by a text encoder. This CLIP text embedding is used as an input by the prior model (diffusion) in stage one to create a CLIP image embedding. The decoder creates the image in stage two using this CLIP image embedding as input.

Diagram

Description automatically generated

***Fig. 4. The architecture of DALL-E 2***

There are two steps in DALL-E 2's training. First, image and text pairs, as well as CLIP text embeddings, are used to train the prior model (diffusion). Image and text pairs, as well as CLIP image embeddings, are used to train the decoder.

# **Data description**

The CUB-200-2011 dataset is a great resource for academics researching birds and their distinctive traits. It has 11,788 high-quality photos of 200 distinct bird species, making it one of the largest and most complete bird collections online. Each picture in the collection is tagged with 312 binary characteristics, 15 part positions, 1 bounding box, and 1 subcategory label. Since a result, it is an wonderful resource for academics interested in bird behavior and morphology, as it contains a plethora of information that may be utilized to better understand these species. One of the most striking features of the CUB-200-2011 dataset is that 80% of the bird photos had object-image size ratios of less than 0.5. This implies that the birds in these photographs are tiny, which may provide difficulties for object recognition systems. To overcome this, the photos are pre-processed by cropping them using the bounding box coordinates. This keeps the bird in the center of the image and enables for more accurate examination of the bird's traits and activities. The CUB-200-2011 dataset has been utilized in a number of research initiatives, including investigations into bird song, avian visual perception, and the influence of habitat degradation on bird populations. The collection has also been utilized by researchers to create new computer vision algorithms for bird detection and identification. Overall, the CUB-200-2011 dataset is an excellent resource for anybody interested in birds and their distinctive traits. Its extensive information and high-quality photographs make it an valuable resource for scholars seeking to learn more about these interesting critters.



Fig. 5. StackGAN result:





# **Results**

#### **StackGAN Results**

Text: “Yellow colored bird with black crown.



#### **StackGAN++ Results**

Text: “Yellow colored bird with black crown.



Text: “A greyish-blue colored bird with a white-grey face, and big blue fee.”

#### **DALL-E Results**

Text: “Yellow colored bird with black crown.”

Text: “A greyish-blue colored bird with a white-grey face, and big blue fee.”

# **Comparison**

***Table I. Inception score metrics***

|  |  |  |
| --- | --- | --- |
|  | **StackGAN** | **StackGAN++** |
| **CUB-200-2011** | 3.66 | 4.60 |
| **Oxford-102** | 3.57 | 4.44 |
| **MSCOCO** | 14.99 | 18.65 |

The quality of the produced pictures was assessed using a variety of measures, including Inception Score (IS) and Fréchet Inception Distance (FID). StackGAN++ regularly beat StackGAN on these criteria, creating pictures with greater quality and more diversified visual information. StackGAN++ can create high-resolution pictures of size 512x512, whereas StackGAN can only generate images of size 256x256. StackGAN++'s increased picture size enables it to create more detailed and visually pleasing images. Because of its bigger size and more complicated design, StackGAN++ requires longer training durations than StackGAN. However, the increased quality and diversity of the generated photos justifies the extended training period. Overall, the trials demonstrate that StackGAN++ outperforms StackGAN in terms of picture quality, variety, and resolution, albeit at the expense of longer training periods. DALL-E2 can create high-resolution photos up to 1024x1024, but DALL-E can only make images up to 256x256 or 512x512. DALL-E2's bigger picture size enables it to create more detailed and visually appealing images. DALL-E2 can generate a larger range of pictures, including objects, animals, and sceneries, whereas DALL-E can only generate things and animals. DALL-E2'sgreaterdiversity is accomplished using a hierarchical generating method that creates pictures in phases. In terms of picture quality, variety, and resolution, the trials reveal that DALL-E2 surpasses DALL-E. DALL-E2'simproveddiversity is particularly amazing, since it can generate a larger range of pictures, including complex settings and objects, which DALL-E could not.

##### **References**

1. Huang, Xun, Yixuan Li, Omid Poursaeed, John Hopcroft, and Serge Belongie. "Stacked generative adversarial networks." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5077-5086. 2017.
2. Ye, Xinchen, Hongcan Xu, Xiang Ji, and Rui Xu. "Underwater image enhancement using stacked generative adversarial networks." In Advances in Multimedia Information Processing–PCM 2018: 19th Pacific-Rim Conference on Multimedia, Hefei, China, September 21-22, 2018, Proceedings, Part III, pp. 514-524. Cham: Springer International Publishing, 2018.
3. Feng, Tuo, and Dongbing Gu. "SGANVO: Unsupervised deep visual odometry and depth estimation with stacked generative adversarial networks." IEEE Robotics and Automation Letters, vol. 4, no. 4, pp. 4431-4437, 2019.
4. Zhang, Han, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N. Metaxas. "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks." In Proceedings of the IEEE international conference on computer vision, pp. 5907-5915. 2017.
5. Zhang, Han, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N. Metaxas. "Stackgan++: Realistic image synthesis with stacked generative adversarial networks." IEEE transactions on pattern analysis and machine intelligence 41, no. 8 (2018): 1947-1962.
6. Tang, Youbao, Jinzheng Cai, Le Lu, Adam P. Harrison, Ke Yan, Jing Xiao, Lin Yang, and Ronald M. Summers. "CT image enhancement using stacked generative adversarial networks and transfer learning for lesion segmentation improvement." In Machine Learning in Medical Imaging: 9th International Workshop, MLMI 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Proceedings 9, pp. 46-54. Springer International Publishing, 2018.
7. Ramesh, Aditya, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. "Hierarchical text-conditional image generation with clip latents." arXiv preprint arXiv:2204.06125 (2022).
8. Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. "High-resolution image synthesis with latent diffusion models." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10684-10695. 2022.
9. Ramesh, Aditya, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. "Zero-shot text-to-image generation." In International Conference on Machine Learning, pp. 8821-8831. PMLR, 2021.
10. Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. "Improving language understanding by generative pre-training." (2018).
11. Nichol, Alex, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. "Glide: Towards photorealistic image generation and editing with text-guided diffusion models." arXiv preprint arXiv:2112.10741 (2021).
12. Doersch, C. (2016). Tutorial on variational autoencoders. arXiv preprint arXiv:1606.05908.
13. Gauthier, J. (2015). Conditional generative adversarial networks for convolutional face generation. In Tech Report.
14. Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1125-1134).
15. Child, R. (2020). Very deep vaes generalize autoregressive models and can outperform them on images. arXiv preprint arXiv:2011.10650.
16. Galatolo, F. A., Cimino, M. G., & Vaglini, G. (2021). Generating images from caption and vice versa via clip-guided generative latent space search. arXiv preprint arXiv:2102.01645.
17. Dhariwal, P., & Nichol, A. (2021). Diffusion models beat gans on image synthesis. Advances in Neural Information Processing Systems, 34, 8780-8794.
18. Koh, J. Y., Baldridge, J., Lee, H., & Yang, Y. (2021). Text-to-image generation grounded by fine-grained user attention. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 237-246).
19. Gao, Y., Liu, J., Xu, Z., Zhang, J., Li, K., Ji, R., & Shen, C. (2022). Pyramidclip: Hierarchical feature alignment for vision-language model pretraining. Advances in Neural Information Processing Systems, 35, 35959-35970.
20. Alayrac, J. B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., ... & Simonyan, K. (2022). Flamingo: a visual language model for few-shot learning. Advances in Neural Information Processing Systems, 35, 23716-23736.

**.**