IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 11, May 2023

Text To Image Synthesis using Generative Adversarial Networks (GAN)

Aniket Kanare[1], Siddhant Kadam[2], Shivam Koshta[3], Prakhar Dadheech[4], Prof. Asmeeta Mali[5]
Students, Department of Engineering[1,2,3,4,]
Assistant Professor, Department of Engineering[5]
Dr. D. Y. Patil Institute of Engineering & Technology, Pimpri, Pune, Maharashtra, India

Abstract: *The project aims to provide a system to synthesize text into a newly built image with the help of Deep Learning algorithms. The process of creating images automatically from supplied text is known as text to image synthesis. However, current AI systems are still far from being able to automatically create realistic images from text, which would be both interesting and useful. To learn discriminative text feature representations, however, recently, general, and potent recurrent neural network designs have been created. For generating images, GAN (Generative Adversarial Network) models are used. Recent progress has been made using Generative Adversarial Networks (GAN). A very good example of deep learning is the transformation of text into images. Deep convolutional generative adversarial networks (GANs) have started to produce incredibly captivating images of particular categories, such faces, album covers, and interiors of rooms. To bridge these developments in text and image modelling, efficiently converting visual notions from characters to pixels, we create a revolutionary deep architecture and GAN formulation in this study*

Keywords: Artificial Intelligence (AI), Generative Adversarial Network (GAN), Machine Learning (ML), Natural Language Tokenizer (NLTK), Python Pickle, Tensor Flow, and Tensor Layer

I. INTRODUCTION

Existing AI systems are still far from being able to automatically create realistic images from text, which would be both interesting and useful. However, to learn discriminative text feature representations, recent years have seen the development of general and strong recurrent neural network designs. Deep convolutional generative adversarial networks (GANs) have started to create incredibly captivating images of categories, such faces, album covers, and interiors of rooms.

As we translate visual notions from characters to pixels in this study, we create a revolutionary deep architecture and GAN formulation to bridge these developments in text and image modelling. We illustrate our model's capacity to create realistic images of birds and flowers from thorough text descriptions.

A. MOTIVATION

The motive of building this project is to accomplish a system that will generate the instance of images with linguistic input so that the human aspect of creativity can be further enhanced using deep learning methods. It implies that a non-living program could replicate human creativity and out-of-the-box thinking.

B. PROBLEM DEFINITION

Creating a simple and effective model to generate images based on detailed visual descriptions. To demonstrate that the model could synthesize many plausible visual interpretations of a text caption.

C. ADVANTAGES & LIMITATIONS

Advantages:

- It can learn the distribution of real samples and explore the real structure of samples.
- It has a more powerful ability for prediction.

- The vulnerability of samples is common in many machine learning models, and GAN is very robust to generate samples.
- The difficulty of small sample machine learning can be alleviated by generating fake and true samples by GAN.
- Compared with reinforcement learning, adversarial learning is closer to the human learning mechanism.

Limitations:

- Although GANs use the adversarial learning concept, model convergence and the presence of an equilibrium point have not yet been proved.
- It is challenging to get quality training outcomes if two adversarial networks are not balanced and synchronized throughout the training process. The training procedure might be unstable, though, because it is difficult to control how the two adversarial networks synchronize
- GANs have the typical flaw (i.e., poor interpretability) of neural networks as they are generative models based on neural networks.
- Despite the diversity of the samples produced by GANs, the collapse mode problem remains. Mode collapse describes situations where the generator creates several pictures with the same color or texture themes, which are hardly distinguishable to humans.

D. APPLICATIONS & USE CASES

GANs have a lot of real-life applications, some of which are:

- Generating examples for image datasets is very handy in medicine or material science, where there’s little data to work with.
- Video game designers can utilize GAN to generate realistic human faces.
- To improve risk management in a firm, GANs may be used to mimic the worst-case scenario.

Other use cases of GAN could be:

- Text-to-Image Translation
- Face Frontal View Generation
- Generate New Human Poses
- Photos to Emojis
- Face Aging
- Super Resolution
- Photo Inpainting
- Clothing Translation
- Video Prediction
- 3D Object Generation

E. LITERATURE REVIEW

[1] Ankit Yadav, Dinesh Kumar Vishwakarma, Recent Developments in Generative Adversarial Networks: A Review (Workshop Paper), 2020.

They created a simple and effective model for picture production after completing a combined analysis of the articles and organizing the project's implementation. They planned to utilize the model on other datasets and improve it in the future to produce images with a high quality.

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

For creating images, they presented the Deep Recurrent Attentive Writer (DRAW) neural network architecture. A unique spatial attention mechanism that replicates the foveation of the human eye and a framework for sequential variational auto-encoding that enables the repeated creation of complex visuals are combined in DRAW networks. The method significantly surpasses the MNIST state-of-the-art for generative models, and when trained on the Street View House Numbers dataset, it produces pictures that are visually indistinguishable from actual data.

[3] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, StackGAN: Text to Photorealistic Image Synthesis with Stacked Generative Adversarial Networks" in Rutgers University and Lehigh University August 2017.

They created high-quality pictures from text descriptions, which is a difficult computer vision issue with numerous real-world applications. The content of the above descriptions can be reflected in the samples produced by existing text-to-image techniques, but they are missing crucial subtleties and vivid object features. They suggested Stacked Generative Adversarial Networks (Stack GAN) in this study to create 256x256photorealistic pictures based on text descriptions. Through a sketch-refinement process, they divided the challenging problem into more manageable subproblems.

[4] Scott Reed, Zeynep Akata, Xinchen Yan, LajanugenLogeswaran, Bernt Schiele, HonglakLee, Generative Adversarial Text to Image Synthesis" in University of Michigan and Max Planc Institute for Informatics June 2016.

In this study, they created a straightforward and efficient technique for creating graphics from thorough visual descriptions. They showed how the model could combine a variety of realistic visual interpretations of a particular text caption. The text to image synthesis on CUB was significantly enhanced by their manifold interpolation regularizer. They demonstrated how to separate style and content, as well as how to move bird poses and backgrounds from query photos onto text descriptions. Finally, using their findings from the MS-COCO dataset, they showed how generalizable their method was for creating pictures with many objects and changing backgrounds. They intended to further expand the model to higher quality photos and incorporate more sorts of text in further work.

II. METHODOLOGY

A method for unsupervised learning called Generative Adversarial Network (GAN) includes creating new instances. Neural networks are used by generative adversarial networks to generate new data instances. It may be used to create voices and images. When a generative model is learned and trained using neural networks, the term "generative adversarial networks" is used. The two components of GAN are the Generator and Discriminator.

Generator: It generates new instances of data, most of which are bogus samples, and sends them to the discriminator in an effort to trick the latter.

Discriminator: It makes a distinction between authentic samples and fake samples produced by the Generator.

Deep neural networks are used in generators and discriminators. The Discriminator's objective is to recognize accurate data, whereas the Generator's objective is to deceive the Discriminator. Both Generator and Discriminator are in opposition to one another. The generator makes every effort to persuade the discriminator that the fake instances it generates are actual samples of data, and it also raises the probability of errors while the discriminator detects the true ones. As a result of several repetitions of these stages, both sub-models learn substantially faster. The Discriminator is first trained on samples of real data to see if it can recognize those samples as real. Again, false data is used to train the discriminator so that it can distinguish between real and fake photos. Based on the Discriminator's findings, the Generator is likewise trained to become better. An extension of the commonly used Generative Adversarial Network is the Deep Convolutional GAN. Since vectors are built up of latent variables in this situation, Generator must create a vector to produce new data. It takes a long time for the GAN model to train.

A. LIBRARIES & FRAMEWORKS

We have employed the Generative Adversarial Network (GAN), a deep learning technique that combines a Generator and a Discriminator. For the creation of text to images, we also used TensorFlow, NumPy, NLTK, and TensorLayer.

TensorFlow

TensorFlow is a machine learning library. In comparison to other deep-learning libraries, it compiles quicker. Additionally, it supports computers that use both GPU and CPU.

NLTK (Natural Language Toolkit) tokenizer

The NLTK (Natural Language Toolkit) tokenizer was used to divide the text into smaller, word-sized pieces. It enables the computer to examine, prepare, and comprehend the written content that the user inputs.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 11, May 2023

TensorLayer
To create different layers in the network for the Generator and Discriminator, such as the input layer, convolutional 2D layer, dense layer, etc., the model is trained using Tensor Layer, a library built on top of TensorFlow.

Python Pickle
The Python Pickle module is used to serialize data, which entails converting objects into bytes so that they may be conveniently stored in files or transported.

B. FUNCTIONAL REQUIREMENTS

Algorithm
We have used the GAN CLS algorithm for training the discriminator and generator.
GAN-CLS Training Algorithm:

- Input - minibatch images, matching text
- Encode matching text description.
- Encode mismatching text description.
- Draw a random noise sample.
- The Generator will pass it to the Discriminator.
- The pairings for training the model and assessing result will be formed in between {actual image, correct text} {actual image, incorrect text}, and {fake image, correct text}.
- Update the Discriminator.
- Update the Generator.

According to the method, a generator will create fake samples and provide them to the discriminator, who will then get three pairs of inputs. Out of the three combinations—right text with the real picture, incorrect text with the real image, and fake image with correct text—the combination of correct text with the real image produces the most accurate results. The Discriminator is trained more precisely using these inputs.

Dataset
The Oxford-102 flower dataset was used. There are 8,192 photos of flowers in Oxford-102, representing all the various kinds. We used 8000 photos to train the model and 189 photographs of flowers to test it. Additionally, 10 captions are considered for each image, making it simple to train and advantageous to generate correct results. This is the project's flowchart, which shows how the process works. The first input is sent to the generator and discriminator in the form of text, and different parameters, including picture resolution, are defined to set up a network.

Graphical User Interface
We have used the Python library PySimpleGUI for the Graphical User Interface (GUI). The theme is especially designed for creating beautiful windows that display the user's creativity while using colors on the GUI. It is simpler to comprehend and put into practice. The project becomes more engaging and approachable by including a GUI. A good GUI encourages the user to interact more and learn the process.

User Interface
A Python toolkit called PySimpleGUI encapsulates tkinter, Qt (pyside2), wxPython, and Remi (for browser compatibility), enabling incredibly quick and easy-to-understand GUI programming. By default, PySimpleGUI uses the tkinter library, but the user may switch to another one by modifying just one line.

C. NON-FUNCTIONAL REQUIREMENTS

Performance Requirements
The application should be portable and possible to users of Windows and Linux with pre-installed python3. The response time for displaying a particular webpage should not be greater than 3-4 seconds for a respectable internet

connection speed. The dataset should be well labelled and wide with minimum of 5000 Images to train the GAN model. Error handling should be implemented, and the application should be able to manage all runtime errors. The application should be flexible for future enhancements.

Software Quality Attributes

Our software has many qualities attribute that are given below:

- Adaptability: This software is adaptable for all users
- Availability: This software is freely available to all users. The availability of the software is easy for everyone.
- Maintainability: After the deployment of the project if any error occurs then the software developer can easily maintain it.
- Reliability: The performance of the software is better which will increase the reliability of the Software.

D. ANALYSIS METHOD

Generative models may produce new data instances. Models that use discrimination distinguish between several types of data items. A discriminative model might distinguish between a dog and a cat, and a generative model could create new images of animals that resemble real animals. One type of generative model is the GAN.

Given a set of labels Y and a set of data instances X, one can say informally: Generative models capture the joint probability p(X, Y) or just p(X) in the absence of labels. The conditional probability p(Y | X) is captured by discriminative models.

A generative model explains the data distribution and the probability of a specific case. Since they can assign a probability to a succession of words, generative models (which are often considerably simpler than GANs) are frequently used to predict the next word in a sequence.

A discriminative model just indicates how probable a label is to apply to an instance, not whether a specific case is likely or not. Keeping in mind that this term is broad. Generative models come in a variety of forms. One type of generative model is the GAN.

E. MODELLING PROBABILITIES

Both types of models are not required to provide back a probability-based number. By replicating that distribution, you may create a model of the data distribution.

A decision tree, for instance, is a discriminative classifier that may classify an instance without giving it a probability. Such a classifier would still be a model as the distribution of every projected label would still correspond to the distribution of labels in the data.

In a similar manner, a generative model may simulate a distribution by generating compelling "fake" data that seems to be taken from the distribution.

F. SYSTEM DESIGN

The system architecture is shown in Fig. 1. has three main components:

- **Generator G:** the generator takes the text description and a random noise z from Gaussian distribution as input and generates a fake image**.**
- **Discriminator D:** the discriminator takes a fake or real image as input along with the corresponding text embedding ψ, then judge whether the input image is fake or real.
- **Captioner C:** the Captioner will take the fake or real image as input and try to decode into a feature vector that matches the size of the text embedding. The Generator and Discriminator can be summarized into a traditional text to image synthesis work using GAN techniques. The Captioner is used to enforce the information flow that maps the images into text descriptions.

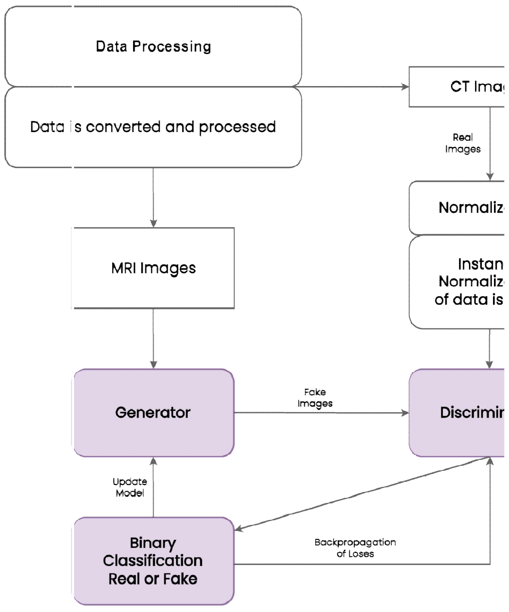

Figure 1: System Architecture

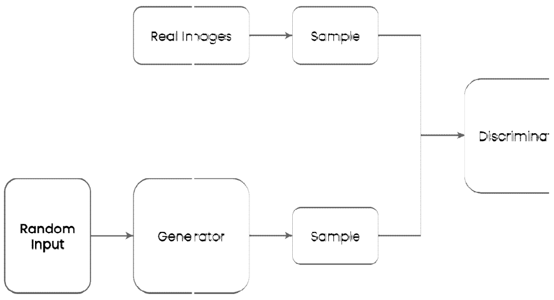

Figure 2: Data Flow Diagram

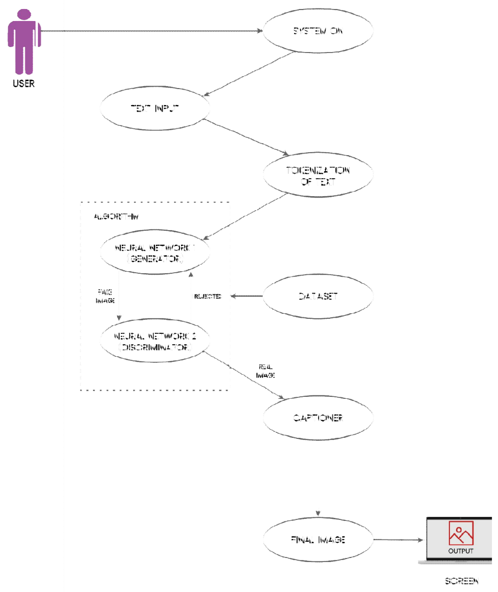

Figure 3: Use Case Diagram

III. ANALYSIS

Generative Adversarial Networks (GAN) is the class of unsupervised learning, which is composed of two models: the generator model G and the discriminator model D. These two models are optimized alternatively. The discriminator D is optimized to distinguish whether an image is a real image or fake image that generated from G, while the generator G is optimized to generate images that are close to the true distribution of the real images such that D cannot distinguish. More concretely, we optimize the following objective function which is like a two-player min-max game:

$$\min_{G}\max_{D} V(D, G) = E_{x\sim Pdata}[\log D(x)] - E_{z\sim Pz}[\log(1-D(G(z)))]$$

where x is sampled from the true data distribution P_{data} and z is a noise vector sampled from Gaussian distribution P_z. It is proved that this min-max game has global optimum when P_g converges to P_{data}

A. INFORMATION FLOW

The task of text to image is to generate the image from the text description and we want to maximize the information of the input text to be expressed in the generated image. The GAN model can generate photo-realistic images, however, we are not sure whether the information from the input text is expressed in the generated image as much as possible, since the information is learned in an implicit way. Also, it is well known that GAN models suffer from mode collapse, which cast further concerns about whether the information is well reserved.

To solve this limitation, we introduce the idea of information flow. That is, given the input text, we want the information from the text to be reserved and expressed as much as possible in the generated image. In other words, we want to maximize the mutual information of the input text and the generated images. However, it is intractable to directly calculate the mutual information of the text and the images. To approximate this problem, we can assume that if the generated image reserves the information from the text, then from the generated image, we should also be able to recover the original text from the generated image. This can be expressed in the chain manner:

$$\text{text} \longrightarrow \text{image} \longrightarrow \text{text}$$

The resulting image can be decoded back to text description or embedding vector in addition to using a text to image pipeline, and this information can be used to maximize the mutual information in an objective function. In this way, we realize combining both pipelines of text to image synthesis and image captioning in the information flow.

B. OBJECTIVE FUNCTIONS

Directly maximizing the mutual information is intractable, instead we approximate this problem by minimizing the Euclidean distance between the text embedding Φ and Φ`. Other approximation methods can be further implemented in future works. The three components in can be optimized alternatively. Specifically speaking, we denote (I_t, t) as the real image and the input sentence pair sampled from the real data distribution P_{data}, with Φ_t as the corresponding text embedding. Following the formulas of GAN, we alternatively train the Generator, the Discriminator and the Captioner by optimizing the following formula:

$$\min_{G,C}\max_{D} L_G + L_D + L_C$$

where L_G, L_D, L_C represent the loss function of the Generator, the Discriminator and the Captioner, respectively. To be exact,

$$L_G = E_{z\sim Pz,t\sim Pdata}[\log(1 - D(G(z, c)),\Phi_t)] + \lambda D_{KL}(N(\mu(\Phi_t), \Sigma(\Phi_t)||N(0, I)))$$

$$L_D = E_{(It,\ t)\sim Pdata}[\log D(It, \Phi_t)] + E_{z\sim Pz,t\sim Pdata}[\log(1 - D(G(z, c)),\Phi_t)]$$

$$L_C = E_{z\sim Pz,t\sim Pdata}[e(C(G(z, c)), \Phi_t)] + E_{(It,\ t)\sim Pdata}[e(C(I_t), \Phi_t)]$$

in which the E function stands for the mean square error between C(G(z, c)) and the text embedding. D_{KL} is the KL divergence regularization term used to enforce the distribution of c to be a normal distribution from VAE. Overall, G is trained to generate fake images from the text description that are closed to the real data distribution, and it should get to the minimum of the Euclidean distance between Φ_t and $C(G(\Phi_t))$. D is trained to distinguish the fake and real images. And C is trained to minimize the Euclidean distance between Φ_t and $C(G(\Phi_t))$ and the distance between Φ_t and $C(I_t)$. In this way, our new network cooperating with three components can learn to reserve and express more mutual information between text descriptions and images.

IV. CONCLUSION

In this study, we created a straightforward and efficient technique for creating graphics from thorough visual descriptions. We showed that the model can combine several realistic visual interpretations of a given text caption. The text to image synthesis on Flowers has been significantly enhanced by our manifold interpolation regularizer. We demonstrated how to separate style and content, as well as how to move bird poses and backgrounds from query photos into text descriptions. With our findings from the Oxford-102 dataset, we finally showed how our method for creating photos with many objects and changing backgrounds is generalizable. The model will be further scaled up to support photos with greater quality and more sorts of text in further development.

REFERENCES

[1] Ankit Yadav, Dinesh Kumar Vishwakarma, Recent Developments in Generative Adversarial Networks: A Review (Workshop Paper), 2020.

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

[3] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, StackGAN: Text to Photorealistic Image Synthesis with Stacked Generative Adversarial Networks" in Rutgers University and Lehigh University August 2017.

[4] Scott Reed, Zeynep Akata, Xinchen Yan, LajanugenLogeswaran, Bernt Schiele, HonglakLee, Generative Adversarial Text to Image Synthesis" in University of Michigan and Max Planc Institute for Informatics June 2016.

[5]Stian Bodnar, Jon Shapiro, Text to Image Synthesis Using Generative Adversarial Networks" in The University of Manchester May 2018.

[6]Tao Xu, Pengchuan Zhang, QiuyuanHuang , Han Zhang, Zhe Gan, Xiaolei Huang, Xiaodong He: Fine- grained text to image generation with attentional generative adversarial networks. CoRR, abs/1711.10485, 2017.

[7] Mehdi Mirza, Simon Osindero, Conditional Generative Adversarial Nets, 2014.