Text to Image Synthesis using Generative Adversarial Network

Atharva Moroney
Courant Inst. of Mathematical Science
New York University
New York, USA
amm9801@nyu.edu

Sachin Malepati
Courant Inst. of Mathematical Science
New York University
New York, USA
sm9449@nyu.edu

Sameer Ahmed
Courant Inst. of Mathematical Science
New York University
New York, USA
sa6142@nyu.edu

***Abstract*—Synthesis of Images using Artificial Intelligence is an interesting and useful task and in recent years, Deep Convolution GANs have produced highly promising results and in turn, gained a lot of popularity for image generation. In this project, we worked on Text to Image Generation and used the paper "Generative Adversarial Text to Image Synthesis" [9] as our primary reference. We implemented the generator and discriminator architecture in DCGAN trained on Oxford 102 Category Flower dataset which enabled it to synthesize photo-realistic images of the flowers from their text descriptions. We also demonstrated the capability of our model to generate such images of flowers from detailed text descriptions.**

1. Introduction

In recent years, research has been done on image synthesis using traditional machine learning algorithms. But the use of Deep Convolutional Generative Adversarial Networks (DCGANs) has improved the results to a great extent.

In this project, we generate synthetic images from a textual description using a Generative Adversarial Network. We will be focusing only on the flowers, so the input text represents the description of the flower. For example, "the petals of this flower are red and the anther are pink".

Previous research work [3] on this subject was approached using attribute representation, essentially using zero-shot vector for visual recognition. Though this yielded good results, it is often difficult to get the attributes as it requires domain-specific knowledge. Whereas, natural language processing can be helpful in providing flexibility in representing objects in visual space. Therefore, we could have the generality of text descriptions with the discriminative power of attributes.

In [10], the deep convolutional neural networks and recurrent neural networks have resulted in high discriminative and generalizable text representations which were better than the zero-shot visual recognition. Similar to this, we plan to learn a mapping directly from words and characters to image pixels.

To achieve the above, we should capture the visual details from the text and then use those details to synthesize a fake image that should be indistinguishable to the human. We can use deep learning capabilities to solve both problems - natural language representation and image synthesis. We try to make use of these two subproblems to achieve our main objective.

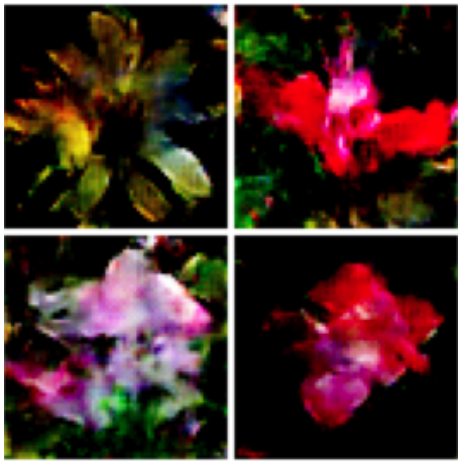

Figure 1. Images generated by our network using the text prompts -
(a)the flower shown has a large pistil and feathery grayish petals as its main feature
(b)this flower is pink in color, and has petals that are curled and wavy
(c)this flower has white and purple petals as well as a white stamen
(d)pink petals with jagged edges with white filament and pink anthers

But there exists one issue here. In deep learning models, there exists multiple possibilities of pixel distribution for a given descriptive text. To address this problem, we can use generative adversarial networks [5] which are optimized to generate the adversarially trained discriminator into predicting that the fake generated images are real. This will help to improve the accuracy of the generative network by trying to outsmart it in every iteration with the use of a discriminator.

Through this project, we aim to mimic the GAN architecture mentioned in the paper which generates fake flower images from a handwritten textual description with a specific training strategy. We will be using the Oxford-102 Flowers dataset along with five text descriptions per image we collected as our evaluation setting. The GAN model will be trained on a smaller subset of categories and we will showcase our results on both training and test data.

2. Related Works

Our model generates images from text and is based on multimodal learning. One of the major challenges that are posed with multimodal learning is shared representation across modalities where the prediction of missing data in one modality is conditioned on another. Numerous research is being conducted in the field of multimodal learning and has shown promising results.

A Deconvolutional network was trained by Dosovitskiy[4] to generate 3D chair renderings conditioned on shape, position, and lighting. It was further enhanced by Yang [12] by including an encoder network and actions which was trained on a recurrent convolutional encoder-decoder and could rotate 3D chair models and human faces conditioned on the sequence of actions. In another research by Reed[11], transformations were encoded from analogy pairs and a convolutional decoder were used to predict shapes, game characters and cars. Convolution decoder networks have also helped the generator network module of GANS. Denton [1] used the laplacian pyramid of generators and discriminators to generate high-resolution images conditioned on class labels. Radford [8] developed an effective and stable image synthesis architecture using a convolutional decoder with batch normalization.

Our project differs from each of the models described above in the sense that we are developing image pixels based on the character of text rather than class labels. Moreover, as in our reference paper, we are using a multifold interpolation regularizer for the GAN generator which immensely improves the quality of the generated images. Most of the work that has been done in the field of multimodal learning relies on target retrieval, however recent development in the field of recurrent neural network decoders to generate text conditioned on images paved the way for working toward using multimodal learning to generate images from text.

In another research to generate images from the text, Mansimov el al used a variational recurrent autoencoder which paints the image in multiple steps. It clearly shows evidence of generating novel images instead of simply memorizing them. Although the results generated by the model are impressive, it is far from realistic. On the other hand, our project is based on GAN and can generate realistic images.

We took a lot of learnings from the previously discussed models to develop our model which generates image pixels based on text characters. We used the Oxford flower-102 dataset embedded with the texts to train and generate images for our model.

3. Background

In this section, we describe some of the previous works on which our model is based.

3.1. Generative Adversarial Networks(GANs)

GANs are a set of two models that competes in a two-player minimax game. The models are termed Generator and Discriminator where the task of the Generator is to generate images so as to fool the Discriminator into believing it to be a real image whereas the task of the Discriminator is to identify the real and generated images. Formally, the Generator(G) and Discriminator(D) plays the following game on V(D, G)

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{x\sim p_z(z)}[\log(1 - D(G(z)))] \quad (1)$$

It has been proved by Goodfellow [6] that the above minimax game has a global optimum when pg = pdata, and given that G and D have enough capacity pg converges to pdata. At the beginning of the training, the samples generated by G are extremely bad and rejected by D with high confidence. It has also been observed that training the model is more efficient when G tries to maximize $\log(D(G(z)))$ instead of minimizing $\log(1 - D(G(z)))$.

3.2. Deep symmetric structured joint embedding

We followed Reed[11] approach of using deep convolutional and recurrent text encoders to learn correspondence function with images which is used to obtain visually-discriminative vector representation of the text descriptions. The following function is optimized to train the text classifier induced by the learned correspondence function

$$\frac{1}{N}\sum_{n=1}^{N} \Delta(y_n, f_v(v_n)) + \Delta(y_n, f_t(t_n)) \quad (2)$$

where $\{(v_n, t_n, y_n) : n = 1, ..., N\}$ is the training data set, Δ is the 0-1 loss, v_n are the images, t_n are the corresponding text descriptions, and y_n are the class labels. Classifiers f_v and f_t are parametrized as follows:

$$f_v(v) = \arg\max_{y\in\mathcal{Y}} \mathbb{E}_{t\sim\mathcal{T}(y)}[\phi(v)^T\varphi(t))] \quad (3)$$

$$f_t(t) = \arg\max_{y\in\mathcal{Y}} \mathbb{E}_{v\sim\mathcal{V}(y)}[\phi(v)^T\varphi(t))] \quad (4)$$

where ϕ is the image encoder, φ is the text encoder (e.g. a character-level CNN or LSTM), $\mathcal{T}(y)$ is the set of text descriptions of class y and likewise $\mathcal{V}(y)$ for images.

The main idea here is that the text encoding belonging to a corresponding image class should have a higher compatibility score.

4. Dataset

We have used the Oxford 102 Category Flower Dataset collected by Maria-Elena Nilsback and Andrew Zisserman. In total, the dataset contains 8,189 images spread across 102 flower categories where the range of the number of images for each class lies between 40 and 258[7]. We have used 82 classes of data for the training and validation set whereas the remaining 20 classes were for the test set.

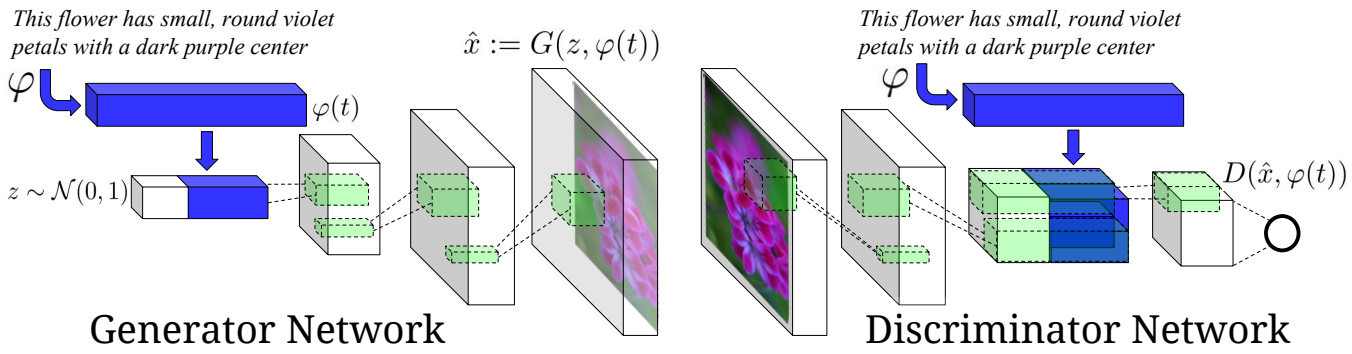

Figure 2. Our GAN architecture

5. Method

In this project, we have used Generative adversarial network conditioned on text features which are encoded by recurrent neural network. Generator as well as the Discriminator perform inference in a feed-forward manner conditioned on the text features.

5.1. Network Architecture

We are using the same notation for the Generator and the Discriminator as in the original paper. The generator network is denoted by $G : \mathbb{R}^Z \times \mathbb{R}^T \rightarrow \mathbb{R}^D$ whereas the discriminator network is denoted by $D : \mathbb{R}^D \times \mathbb{R}^T \rightarrow \{0, 1\}$, where D is the dimension of the image and T is the dimension of the text embedding. The image of the network architecture can be found in Figure 2.

In the Generator, the noise prior is sampled from $z \in \mathbb{R}^Z \sim \mathcal{N}(0, 1)$ and the text is encoded using the encoder φ. The dimension of the text encoding is reduced to 128 using fully-connected layer followed by Leaky ReLU, the result is then concatenated to the noise vector z. Thereafter the concatenated noise vector is feed-forward through the G to generate an image using $\hat{x} \leftarrow G(z, \varphi(t))$. The image generated by the Generator G is conditioned on the embedded text description φ.

In the Discriminator too, we reduce the dimension of the text embedding using a fully connected layer, and several levels of stride 2 convolutions are performed with batch normalization followed by leaky ReLU. When the spatial dimension is reduced to 4*4 we concatenate and perform a depth concatenation with the text embeddings. Finally, a 4*4 convolution is used to compute the final score of the Discriminator. It should be noted that Batch Normalization is performed at all convolution layers.

6. Experiment and Results

In this section, we share the experimental details and present the results that we obtained from our GAN model. As mentioned in the above Dataset section, we have split our dataset in training+validation and test sets. We have also used a pre-trained word2vec model that generates text embeddings for each of the image in our dataset. We have used a size of 64*64*3 for each of the images in the training set. For the generator as well as the discriminator we concatenate the 128-dimensional vector prior to convolution. Some of the hyperparameters include a base learning rate of 0.0002 with 0.5 momentum for the ADAM optimizer. The noise for the generator was sampled from a 100 dimensional vector trained for 100 epochs on a batch size of 64.

We can see the losses for the generator and the discriminator over a period of 100 epochs in Figure 5. It can be observed that the losses for the generator and the discriminator converge after reaching the Nash equilibrium. The final results of the images are generated after the 100th epoch which can be clearly seen in Figure 3. The quality of images generated by the model is impressive with bright and natural colors and looks almost the same as the real images from the dataset in Figure 3

Algorithm 1 Minibatch stochastic gradient descent training of GANs. Let k be the number of steps that are applied to the Discriminator D. We have used $k = 1$ for ease.

for $iter = 1$ **to** $number\ of\ iterations$ **do**
  for $i = 1$ **to** k **do**
    - Sample a minibatch of m noise samples $z(1), ..., z(m)$ from the noise prior $p_g(z)$
    - Sample a minibatch of m samples $d(1), ..., d(m)$ from the data generating distribution $p_{data}(d)$
    - Ascend the stochastic gradient to update the discriminator.
  end
    - Sample a minibatch of m noise samples $z(1), ..., z(m)$ from the noise prior $p_g(z)$
    - Descend the stochastic gradient to update the generator
end

Figure 3. Real Images from the dataset

Figure 4. Generated Images by our network

The complete code for our implementation can be found here https://github.com/ath-08/Generative-Adversarial-Text-to-Image-Synthesis-PyTorch.

7. Conclusion

As part of this project, we developed a GAN model using a Generator and a Discriminator where both play a minimax game to outsmart the other. The task of the Generator is to generate fake images based on the text description and the task of the Discriminator is to identify if the image is a real image or generated image based on the text embedding. Our

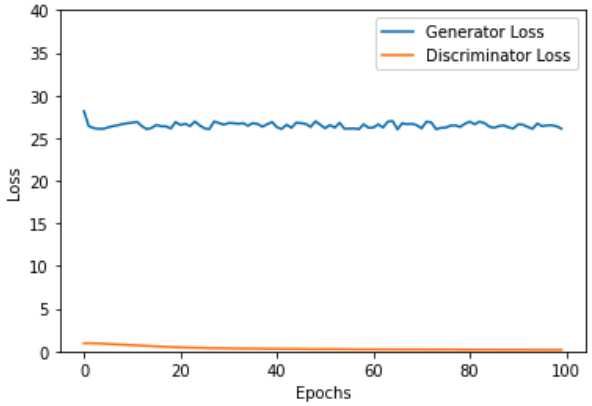

Figure 5. Loss plot

model converges well on the oxford-102 flower dataset that we used and has been able to produce realistic results as shown in Figure 4. Generally, it is considered that the GANs take a lot of training time and are difficult to converge, but as we have shown here, if we use appropriate hyperparameters we can converge the model well and produce astonishing results.

8. Future Scope

Although we have trained our model on Oxford 102 flower dataset, owing to the generalization of the concept of the Generator and the Discriminator used in the GAN, the same model can be used to train more diverse datasets. One of the more popular datasets like MS-COCO which has a large number of images with a diverse set of classes can be used with text embedding to train this model and enhance its generative capability to more diverse sets of outputs. Although we have obtained high-quality results using the DCGAN but there is a growing popularity of the thermodynamics-based diffusion model for image generation. It has shown significant improvement in image quality and training convergence over the GAN models. [2]. This task of text-to-image generation can be extended and implemented using the diffusion model and a comparison can be done between the GAN-based model and diffusion model for the quality of images.

Acknowledgments

We would like to thank Professor Rob Fergus for guiding us throughout the project and sharing his valuable suggestions and inputs which have helped us build and visualize this project. We would also thank the Courant Institute of Mathematical Sciences, NYU for allocating us the GPU resources to train our model.

References

[1] Emily L Denton et al. "Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks". In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes et al. Vol. 28. Curran Associates, Inc., 2015. URL: https://proceedings.neurips.cc/paper/2015/file/aa169b49b583a2b5af89203c2b78c67c-Paper.pdf.

[2] Prafulla Dhariwal and Alex Nichol. *Diffusion Models Beat GANs on Image Synthesis*. 2021. DOI: 10.48550/ARXIV.2105.05233. URL: https://arxiv.org/abs/2105.05233.

[3] Ali Farhadi et al. "Describing objects by their attributes". In: *2009 IEEE conference on computer vision and pattern recognition*. IEEE. 2009, pp. 1778–1785.

[4] Philipp Fischer et al. *FlowNet: Learning Optical Flow with Convolutional Networks*. 2015. DOI: 10.48550/ARXIV.1504.06852. URL: https://arxiv.org/abs/1504.06852.

[5] Ian Goodfellow et al. "Generative adversarial networks". In: *Communications of the ACM* 63.11 (2020), pp. 139–144.

[6] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 2014. DOI: 10.48550/ARXIV.1406.2661. URL: https://arxiv.org/abs/1406.2661.

[7] Maria-Elena Nilsback and Andrew Zisserman. "Automated Flower Classification over a Large Number of Classes". In: *2008 Sixth Indian Conference on Computer Vision, Graphics Image Processing*. 2008, pp. 722–729. DOI: 10.1109/ICVGIP.2008.47.

[8] Alec Radford, Luke Metz, and Soumith Chintala. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. 2015. DOI: 10.48550/ARXIV.1511.06434. URL: https://arxiv.org/abs/1511.06434.

[9] Scott Reed et al. "Generative adversarial text to image synthesis". In: *International conference on machine learning*. PMLR. 2016, pp. 1060–1069.

[10] Scott Reed et al. "Learning deep representations of fine-grained visual descriptions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 49–58.

[11] Scott E Reed et al. "Deep Visual Analogy-Making". In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes et al. Vol. 28. Curran Associates, Inc., 2015. URL: https://proceedings.neurips.cc/paper/2015/file/e07413354875be01a996dc560274708e-Paper.pdf.

[12] Jimei Yang et al. *Weakly-supervised Disentangling with Recurrent Transformations for 3D View Synthesis*. 2016. DOI: 10.48550/ARXIV.1601.00706. URL: https://arxiv.org/abs/1601.00706.