Text to Image Synthesizer using Deep Convolutional GANs

Harsh Dubey
Courant Institute of Mathematical Sciences
New York University
New York, NY 10011
hd2225@nyu.edu

Ayush Jain
Courant Institute of Mathematical Sciences
New York University
New York, NY 10011
aj3152@nyu.edu

Abstract

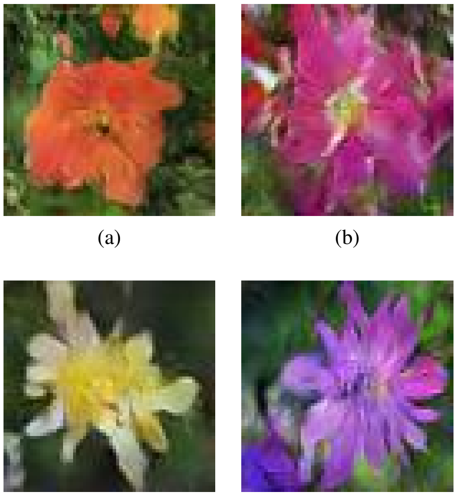
One of the interesting applications of the Computer Vision is to generate realistic images from the textual descriptions. While traditional machine learning algorithms were successful in past to generate good synthetic images from input text data, the advent of Deep Convolutional Generative Adversarial Networks (DCGANs) revolutionized the field. Traditional machine learning models were unable to capture the necessary detail and randomness in the synthesized images. DCGANs bridged these gaps by learning from large datasets to include high-frequency details and features in the images synthesized. In this project, we discuss and implement the generator and discriminator architecture in DCGAN trained on Oxford 102 Category Flower dataset which enables it to synthesize more photo-realistic images of the flowers from its text descriptions. DCGANs finds its vast application via image synthesis and style transfer in the animation, photo-editing and other domains.

1. Introduction

In this project, we implemented Deep Convolutional Generative Adversarial Networks (DCGANs) which takes single sentence textual descriptions describing the details of the flower like color and structure to synthesize images of flowers. As an instance, the textual descriptions can be described as ”this flower is a beautiful color orange with overlapping petals” or ”the petals on this flower are pink with white stamen”. The traditional machine learning algorithms involved having the visual attributes of the subject encoded in a vector to facilitate zero shot detector and conditional image generation.

While the conventional machine learning algorithms were helpful to provide the strong generalization and good capability to distinguish the key attributes [1], though they are not always easy to work with due to their prerequisite of the domain related knowledge. When we compare it to the natural language processing, it provides us with better flexibility to represent the features generally for all visual possibilities. We are aiming here to have the distinguishing capability of the visual attributes with the flexibility of the textual data.

The advancements in the field of deep convolutional networks have led to the strong distinguishing and generality from the textual representations derived from characters and words. The results from these algorithms are better than zero-shot detector on Caltech-UCSD birds dataset. As an extension to that, in this project we aim to implement transalate textual descriptions in the form of words and characters directly to image pixels.



(a) (b)

(c) (d)

Figure 1: (a) This flower is a beautiful color orange with overlapping petals (b) the petals on this flower are pink with white stamen (c) wide and flat green, white and yellow flower with spiked petals (d) The petals on this flower are purple with purple stamen

In Figure 1. we have synthesized images for corresponding text descriptions from Oxford-102 Flowers dataset.

There are two major components which needs to be addressed while solving this interesting problem. Firstly, we have to figure out a way to get the textual features which can sufficiently represent the key visual details. Secondly, we have to utilize these features to create a photo-realistic fake images which looks indistinguishable from real images. Due to tremendous research in deep learning, we have been able to solve both sub-problems by natural language processing and image synthesis respectively which gives us the opportunity to solve our problem by utilising the tools at hand.

There is a problem which needs to be addressed in standalone deep learning models though which arise due to multiple possibilities of the pixel distribution corresponding to a given textual data. Due to this multi-modality, it gets difficult for just deep learning models to arrive at a specific pixel distribution for given text input. It had been an important issue and discussed extensively in computer vision research. A plausible solution as suggested earlier [14] is to have generative and discriminator networks by conditioning them on same information. Here, the generative network is optimized to deceive the adversarially trained discriminator to convince it to believe that the fake synthesized images are real. Essentially, discriminator helps to improve the accuracy of generative network by trying to outsmart it in every iteration.

In this project, we will be analysing the architecture of the effective Deep Convolutional Generative Adversarial Networks and train it on the manually fed textual descriptions with the corresponding flower images from Oxford-102 Flowers dataset to create photo-realistic synthetic images. We trained our model on a set of training categories and have analyzed and reported the performance of the model on both training and testing datasets.

2. Related Work

There have been great research work carried out in the field of image synthesis focused mainly on learning more about challenges with the multimodal learning. It is also a challenge to predict the missing data on the shared representation across multiple modalities conditioned on one another. There have been developments based on stacked multimodal autoencoder on both audio and video signals which was successful on learning modality invariant solutions [13]. Another conditional prediction model was created to find one modality from another.

Deep convolutional decoder networks also played a great role in synthesizing the photo-realisitic images. Deconvolutional network was trained by Dosovitskiy et al. [6] to generate 3D chair renderings conditioned on various parameters like lighting, shape and size of the objects. In another experiment, an encoder network and actions were also utilized by training the recurrent convolutional neural network on a set of action sequences created from various rotations. In another instance, an convolutional decoder is used to predict visual similarities on various parameters after encoding transformations from analogous pairs.

Generative network module also finds its application in convolutional decoders to implement generative adversarial networks [10]. Images at high resolution was created using Laplacian pyramid of adversarial generator and discriminator while controlling the class labels [4].

In this project, the DCGAN implmenetation takes the input as characters and provides the result in pixels which was one of the first end to end implementation of its kind [14]. They also introduced manifold interpolation regularizer for the GAN generator that dramatically improves the image quality. Recent breakthroughs in recurrent neural network decoders helped to create text descriptions for images (Vinyals et al., 2015; Mao et al., 2015; Karpathy and Li, 2015; Donahue et al., 2015)). Sequential models have been applied to both text in books and movies together to enhance the knowledge base.

Variational recurrent autoencoders have created the images from the input text captions trying to paint the image in multiple steps [15]. The model does great even while synthesizing completely new images which was unknown to the model. Though the results are impressive, they cannot be mistaken for real images, i.e. they cannot be disguised as a human drawn painting.

These past works plays a key role in helping us to implement an effective text base image synthesizer using character-level text encoding and class-conditional Generative Adversarial networks. The image dataset is derived completely from Oxford-102 Flowers dataset with images having their corresponding embedding generated from a natural language processing.

3. Background

We will shed some light on the past relevant research which laid the foundation of our model in this section.

3.1. Generative adversarial networks

There are two major components of any Generative adversarial networks (GANs), namely generator G and a discriminator D which are players of a two player min-max optimization game. The generator creates fake images from random noise distribution and tries to fool discriminator while discriminator tries to identify the synthetic images from real training data. Let, the game played by D and G on V(D,G) as:

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

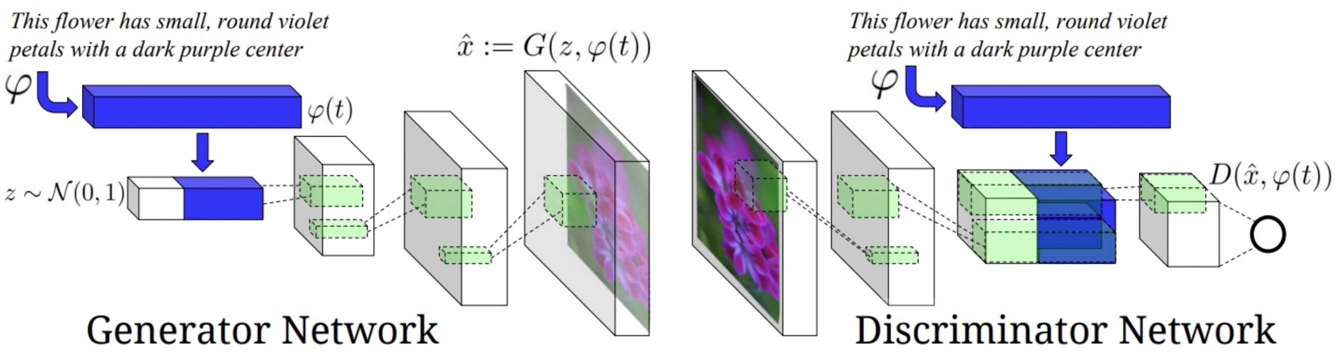

This flower has small, round violet petals with a dark purple center
φ
$\varphi(t)$
$\hat{x} := G(z, \varphi(t))$
$z \sim \mathcal{N}(0,1)$
Generator Network
This flower has small, round violet petals with a dark purple center
φ
$D(\hat{x}, \varphi(t))$
Discriminator Network

Figure 2: Text-conditional convolutional GAN architecture. Text encoding $\varphi(t)$ is used by both generator and discriminator. It is projected to a lower-dimensions and depth concatenated with image feature maps for further stages of processing.

Goodfellow et al. proved in 2014 that the min-max game has global optimum precisely when $p_g = p_{data}$ under milder conditions. In practice, it is best for generator to maximize $\log D(G(z))$ rather than minimizing $\log(1 - D(G(z)))$.

3.2. Deep symmetric structured joint embedding

We can utilize deep convolutional and recurrent text encoders to get a visually-discriminative vector representation of text descriptions as suggested by Reed et al.[14] by learning correspondence function with images. The model is trained by optimizing the following structured loss:

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

4. Method

Our approach entails training a deep convolutional generative adversarial network (DCGAN) on the text features encoded by a hybrid character-level convolutional-recurrent neural network. Both generator network G and the discriminator D feeds the network forward on text features.

4.1. Network architecture

In the network architecture followed we are using the notations as shared here. The generator network is denoted by $G : \mathbb{R}^Z \times \mathbb{R}^T \rightarrow \mathbb{R}^D$ and by discriminator as $D : \mathbb{R}^D \times \mathbb{R}^T \rightarrow \{0, 1\}$. Here, D is the dimension of the image, T is the dimension of text description embedding and Z is the dimension of the noise input to G. Figure 2. represents the network architecture.

Firstly, for the generator, we sample from the noise $z \in \mathbb{R}^Z$~$N(0, 1)$ and then we encode the text query t utilizing the text encoder φ. In the beginning, the text encoding $\varphi(t)$ is compressed utilizing a fully-connected layer to smaller dimension followed by leaky-ReLU and then eventually concatenated to the noise vector z. After this, the inference continues as expected in a deconvolutional network while feeding forward it through the generator G; creating a synthetic image $\hat{a}$.

In the second half, we have discriminator D, which perform performs multiple layers of convolution having stride 2 with spatial batch normalization followed by leaky ReLU. We continue to reduce the dimensionality of the text embedding $\varphi(t)$ in a fully connected layer continued by rectification. It is to be noticed that batch normalisation is conducted on all convolutional layers.

Algorithm 1: Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used k = 1, the least expensive option, in our experiments.

```
for for number of training iterations do
    for k steps do
      • Sample minibatch of m noise samples $\{z^{(1)}, ..., z^{(m)}\}$
        from noise prior $p_g(z)$.
      • Sample minibatch of m examples $\{x^{(1)}, ..., x^{(m)}\}$ from
        data generating distribution $p_{data}(x)$.
      • Update the discriminator by ascending its stochastic
        gradient:
        $\mathbb{E}_{x-P_{data(x)}}[\log D(x)] + \mathbb{E}_{x-P_{data(x)}}[\log(1 - D(G(z)))]$
    end for
    • Sample minibatch of m noise samples $\{z^{(1)}, ..., z^{(m)}\}$
      from noise prior $p_g(z)$.
    • Update generator by descending its stochastic gradient:
      $\mathbb{E}_{x-P_{data(x)}}[\log(1 - D(G(z)))]$
end for
```

The gradient-based updates can use any standard gradient-based learning. We used momentum in our experiments.

Algorithm 1 explains the step by step processing of the

DCGAN architecture implemented in this project in adequate detail.

5. Experiments

In this section, we are analyzing and reporting the results obtained on the Oxford-102 dataset of flower images. The Oxford-102 contains 8,189 images of flowers from 102 different categories having 40745 data points in total.

We split the Oxford-102 flower dataset into training, validation and test data. Oxford-102 has 82 training and validation classes along with 20 test classes. Validation data has 5780 and test data 5575 data points. Training data has having 29390 data points. Corresponding to each image in the dataset, we are utilizing a pre-trained embeddings received from a Word2vec natural language processing algorithm. Having pretrained embeddings for the images increases the speed of training and provides flexibility for more experimentation.

The training image size was maintained at 64 x 64 x 3. The text embedding was projected to 128 dimensional vector for both generator and discriminator before we depth concatenate it into convolutional feature maps.

We used 0.0002 base learning rate and utilized for ADAM solver with 0.5 momentum.The noise from generator was sampled from a 100-dimensional normal distribution while having minibatch size of 64 which is trained for 200 epochs.

The complete implementation and the corresponding code can be found at the following repository: `https://github.com/hardy30894/Text_To_Image_Synthesis_Using_DCGAN`

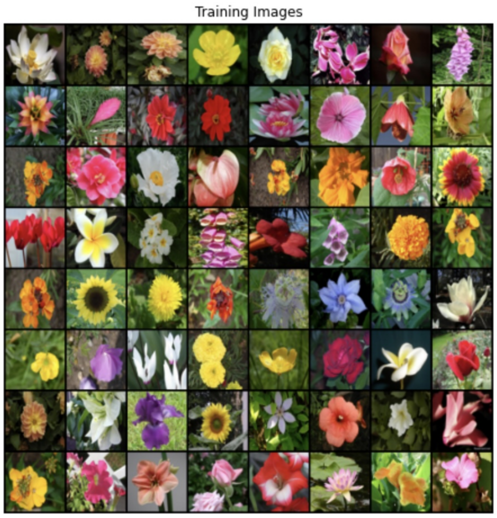


Figure 3: Input training images from Oxford-102 dataset

5.1. Qualitative results

In this section, we will analyze and discuss the results reported after the implementation of DCGANs on the Oxford-102 Flowers dataset as per the design discussed in the last section. In this architecture, generator and discriminator are trying to outsmart each other. Generator synthesizes fake images similar to real images and passes it to discriminator along with input training data sampled randomly. Now discriminator has to predict if it's a fake synthesized image or a real image from training dataset. In the beginning, since generator generates random images from sampled noise, discriminator shoots down all the fake synthesized images generated by generator. This feedback is sent to generator which gets better eventually at generating more photo-realistic images closer to reality. Both discriminator are thereby said to be playing the min-max game here which eventually touches their Nash equilibrium at a certain point where both generator and discriminator losses converge.

In the DCGAN model we have implemented, the generator and discriminator losses at epoch 0 are 1.5 and 40 respectively which falls down to 1 and 25 at 10th epoch respectively.

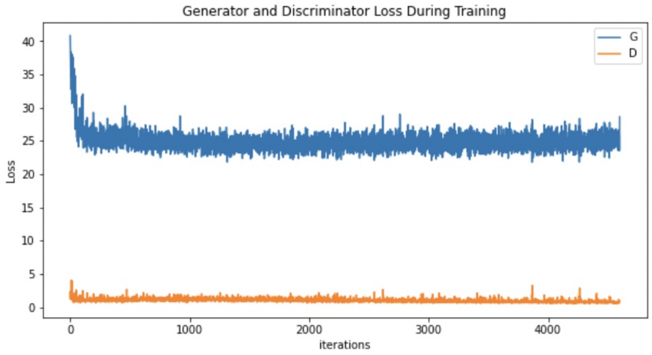


Figure 4: Generator and Discriminator loss over 200 epochs

In Figure 4 we can clearly see the point where the generator and discriminator losses converge after saturating at their Nash equilibrium. Figure 5 shows us the fake images synthesized by the generator at the end of the 200th epoch. We can clearly observe the improvement in the quality of fake images synthesized after training the DCGAN for 200 epochs. The incredible accuracy, colour and structure is the result of the min-max optimization game as a result of the designed generator and discriminator design.

6. Conclusions

In this project, we implemented a simple Deep Convolutional Generative Adversarial Network architecture involving a generator and a discriminator playing a min-max optimization game to improve their losses by trying to outsmart each other. The quality of image synthesized is seen

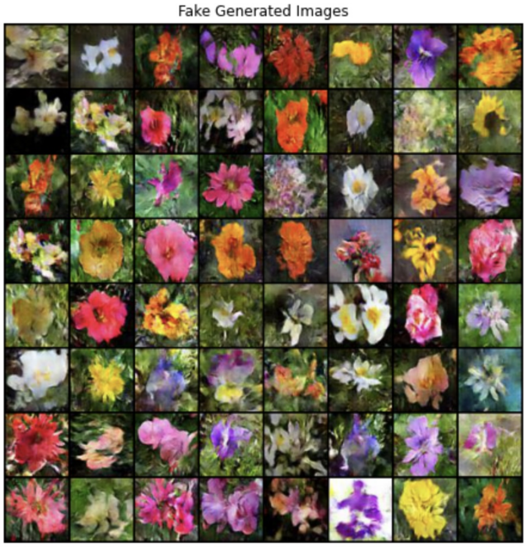


Figure 5: Fake Images Generated on Testing data

to be strongly correlated with the total number of epochs on which the DCGAN is trained. Eventually, we observed that the generator synthesized images of expected color and shape as suggested by input textual data in great detail.

7. Future Scope

We can aim to generate higher resolution images and generalize our approach to include images of multiple objects with varying backgrounds.

In order to improve the image resolution, we can utilise the Stack-GAN architecture and using a more generic image dataset having more diverse set of images like MS-COCO dataset.

7.1. StackGAN

StackGAN aims at creating high resolution fake synthesized images which looks indistinguishable from the real images. In order to achieve that it proposes an architecture which consists of two stage GAN. The two stages are as follows:

- **Stage-I GAN:** In this stage, it just aims to form a basic layout and colors of the object from the textual descriptions along with its background layout derived from a random noise vector is drawn generating a low resolution image.

- **Stage-II GAN:** In this stage, it aims to bridge the gaps of the low resolution image generated in Stage-I GAN by correcting the details and colors of the image derived from again reading the textual data. It helps to give a last finishing touch to the low resolution image, thereby creating a high resolution photo-realistic image.

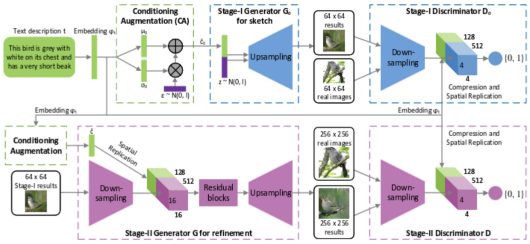


Figure 6: StackGAN Network Architecture

StackGAN++ can also be utilised to get the high resolution images. It is an extension of the existing StackGAN network architecture which is a multi-stage GAN designed in a tree like structure. In this case, the architecture generates mutiple images at mutiple scales for the a given scene which helps StackGAN++ to outperform all existing algorithms in the field.

7.2. Beyond flowers

In this project, we trained a Deep Convolutional Generative Adversarial Generative Network to synthesize fake synthetic images trained on Oxford-102 Flowers dataset. Though, the model is capable of handling the flexibility owing to its generator-discriminator model which is independent of any specific type of data. Since, the loss optimization in this architecture is unlike mean square loss optimization which promotes binding similar features together based on their Eucledian distances, DCGANs doesn't bind similar features on this underlying concept which gives it the flexibility to have more variance in its results. Therefore, the same architecture can be trained on a more diverse dataset like MS-COCO having much more class of objects to synthesize photo-realistic images of multiple objects with varying backgrounds.

Acknowledgements

We would like to thank Prof. Rob Fergus for helping us to visualize DCGANs and guiding us post lectures to resolve the roadblocks. This project was carried out as a part of the curriculum for Computer Vision course at NYU Courant Institute of Mathematical Sciences.

References

[1] Akata, Z., Reed, S., Walter, D., Lee, H., and Schiele, B. Evaluation of Output Embeddings for Fine-Grained Image Classification. In CVPR, 2015.

[2] Ba, J. and Kingma, D. Adam: A method for stochastic optimization. In ICLR, 2015.

[3] Bengio, Y., Mesnil, G., Dauphin, Y., and Rifai, S. Better mixing via deep representations. In ICML, 2013

[4] Denton, E. L., Chintala, S., Fergus, R., et al. Deep generative image models using a laplacian pyramid of adversarial networks. In NIPS, 2015.

[5] Donahue, J., Hendricks, L. A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., and Darrell, T. Longterm recurrent convolutional networks for visual recognition and description. In CVPR, 2015.

[6] Dosovitskiy, A., Tobias Springenberg, J., and Brox, T. Learning to generate chairs with convolutional neural networks. In CVPR, 2015.

[7] Farhadi, A., Endres, I., Hoiem, D., and Forsyth, D. Describing objects by their attributes. In CVPR, 2009.

[8] Fu, Y., Hospedales, T. M., Xiang, T., Fu, Z., and Gong, S. Transductive multi-view embedding for zero-shot recognition and annotation. In ECCV, 2014.

[9] Gauthier, J. Conditional generative adversarial nets for convolutional face generation. Technical report, 2015.

[10] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In NIPS, 2014.

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

[12] Hochreiter, S. and Schmidhuber, J. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

[13] Jiquan Ngiam1 Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, Andrew Y. Ng, Multimodal Deep Learning, 2011

[14] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text-to-image synthesis. In Proceedings of The 33rd International Conference on Machine Learning, 2016

[15] Mansimov, E., Parisotto, E., Ba, J. L., and Salakhutdinov, R. Generating images from captions with attention. ICLR, 2016.