

*A Minor Project Report*

*On*

# **Text to Image generation using Generative Adversarial Networks (GANs)**

*Carried out as part of the course CS1634 Submitted by*

**Vimal Subbiah**

**189301026**

**VI-CSE D**

*In partial fulfilment for the award of the degree*

*of*

**BACHELOR OF TECHNOLOGY**

**In**

**Computer Science and Engineering**



**MANIPAL UNIVERSITY  
JAIPUR**

**Department of Computer Science and Engineering**

**School of Computing and Information Technology**

**Manipal University Jaipur**

***May 2021***

# **ABSTRACT**

Generative adversarial networks (GANs) can generate realistic-looking images that adhere to characteristics described in a textual manner, e.g., an image caption. For this, most networks are conditioned on an embedding of the textual description. Often, the textual description is used on multiple levels of resolution, e.g. first to obtain a coarse layout of the image at lower levels and then to improve the details of the image on higher resolutions. This approach has led to good results on simple, well-structured data sets containing a specific class of objects (e.g., faces, birds, or flowers) at the image centre. Generative modelling is an unsupervised learning task in machine learning that involves automatically discovering and learning the regularities or patterns in input data in such a way that the model can be used to generate or output new examples that plausibly could have been drawn from the original dataset.

Moreover to decompose the hard problem into more manageable sub-problems through a sketch-refinement process. The Stage-I GAN sketches the primitive shape and colors of the object based on the given text description, yielding Stage-I low-resolution images. The Stage-II GAN takes Stage-I results and text descriptions as inputs, and generates high-resolution images with photo-realistic details. It is able to rectify defects in Stage-I results and add compelling details with the refinement process

The main objective of this project is to introduce a novel GAN architecture that specifically models individual objects based on some textual image description. This is achieved by adding object pathways to both the generator and discriminator which learn features for individual objects at different resolutions and scales. Our experiments show that this consistently improves the baseline architecture based on quantitative and qualitative evaluations.

In the end the main goal is to demonstrate plausible visual interpretations of a given text caption using manifold interpolation regularizer substantially improved the text to image synthesis on CUB , show transfer from query images onto text descriptions. Moreover, generate images with multiple objects and variable back- grounds with our results on MS-COCO dataset. In future work, I aim to further scale up the model to higher resolution images and add more types of text.

Finally improve accuracy and use an evaluation metric on text-to-image models to provide more detailed information about how well they perform for different object classes or image captions and how well they align with human evaluation and demonstrate the power of GANs.

# **TABLE OF CONTENTS**

<b><u>1. INTRODUCTION.....</u></b>	<b><u>4</u></b>
1.1. MOTIVATION .....	4
<b><u>2. LITERATURE REVIEW .....</u></b>	<b><u>5</u></b>
2.1. DATASETS .....	5
2.1. DCGANS T2I.....	5
2.3. STACKGAN.....	5
2.3.1. STACKGAN (STAGE-1) .....	5
2.3.2. STACKGAN (STAGE-2) .....	6
2.4. OUTCOME OF LITERATURE REVIEW .....	6
2.5. PROBLEM STATEMENT .....	6
2.6. RESEARCH OBJECTIVES .....	7
<b><u>3. METHODOLOGY AND FRAMEWORK.....</u></b>	<b><u>7</u></b>
3.1. SYSTEM ARCHITECTURE .....	7
3.1.1. DC-GAN .....	7
3.1.2. StackGAN .....	8
3.2. ALGORITHMS .....	8
Algorithms used .....	9
3.2.1. Stage 1 StackGAN .....	9
3.2.2. Stage 2 StackGAN .....	10
3.3. DESIGN METHODOLOGIES .....	11
<b><u>4. WORK DONE.....</u></b>	<b><u>13</u></b>
4.1. IMAGES GENERATED BY DIFFERENT NETWORKS.....	13
4.1.1. DC-GAN .....	13
4.1.2. StackGAN .....	13
4.2. SUMMARY .....	14
4.3. RESULTS & FINAL SCORES.....	14
<b><u>5. CONCLUSION &amp; FUTURE PLAN.....</u></b>	<b><u>15</u></b>
<b><u>6. REFERENCES.....</u></b>	<b><u>16</u></b>

# **1. Introduction:**

## **1.1. Motivation**

The motivation for doing this project was primarily an interest in undertaking a challenging project in an interesting area of research that is in generative networks, the idea of working on a project with a learning curve is always an interesting concept for me. Moreover, working in the ever extending field of AI and Machine learning with the use of GANs which are an exciting and rapidly changing field, delivering on the promise of generative models in their ability to generate realistic examples across a range of problem domains, most notably in image-to-image translation tasks such as translating photos of summer to winter or day to night, and in generating photorealistic photos of objects, scenes, and people that even humans cannot tell are fake.

This problem is more severe because the image resolution increases the GANs only succeeded in generating plausible  $64 \times 64$  images conditioned on text descriptions which usually lack details and vivid object parts, e.g., beaks and eyes of birds. Moreover, they were unable to synthesize higher resolution (e.g.,  $128 \times 128$  &  $256 \times 256$ ) images without providing additional annotations of objects.

Once images and textual descriptions become more complex, e.g., by containing quite one object and having a large variety in backgrounds and scenery settings, the image quality drops drastically. This is often likely because, until recently, almost all approaches only condition on an embedding of the complete textual description, without listening to individual objects. Recent approaches have begun to tackle this by either counting on specific scene layouts or by explicitly that specialize in individual objects. During this work, trying to extend this approach by additionally focusing specifically on salient objects within the generated image. However, generating complex scenes containing multiple objects from a variety of classes remains a challenging problem.

So solving this problem would provide a new approach to image generation and help in various fields.

## 2. Literature Review

### 2.1. Datasets

Dataset	CUB [8]		Oxford-102 [9]		COCO [10]		ImageNet [11]	
	Train	Test	Train	Test	Train	Test	Dog	cat
#Samples	8,855	2,933	7,034	1,155	80,000	40,000	147,873	6500

TABLE 1: : Statistics of datasets. We do not split ImageNet because it is utilized for the unconditional tasks.

### 2.2. DC-GANs T2I

(Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., & Lee, H. (2016, June). Generative adversarial text to image synthesis. In *International Conference on Machine Learning* (pp. 1060-1069). PMLR.) [3]

This is the basic approach is to train a deep convolutional generative adversarial network (DC-GAN) conditioned on text features encoded by a hybrid character-level convolutional recurrent neural network. Both the generator network  $G$  and the discriminator network  $D$  perform feed-forward inference conditioned on the text feature. Generator as  $G : \mathbb{R}^Z \times \mathbb{R}^T \rightarrow \mathbb{R}^D$ , the discriminator as  $D : \mathbb{R}^D \times \mathbb{R}^T \rightarrow \{0, 1\}$ ,

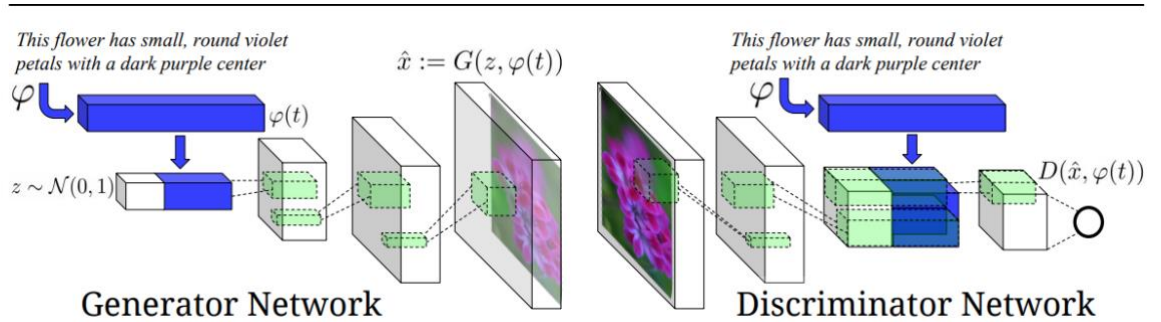


Figure 2.1: DCGAN [3]

### 2.3. StackGAN

#### 2.3.1. StackGAN (Stage-1)

This sketches the primitive shape and basic colors of the object conditioned on the given text description, and draws the background layout from a random noise vector, yielding a low-resolution image.

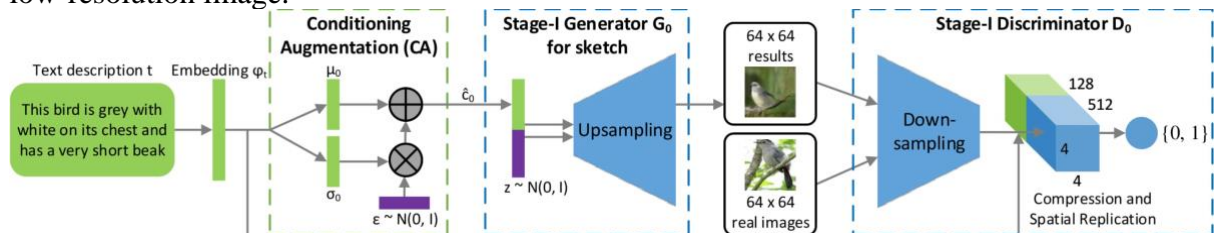


Figure 2.2.1 Stage 1 StackGAN [7]

### 2.3.2. StackGAN (Stage-2)

It corrects defects in the low-resolution image from Stage-I and completes details of the object by reading the text description again(text embed is inputted again), producing a high-resolution photo-realistic image.

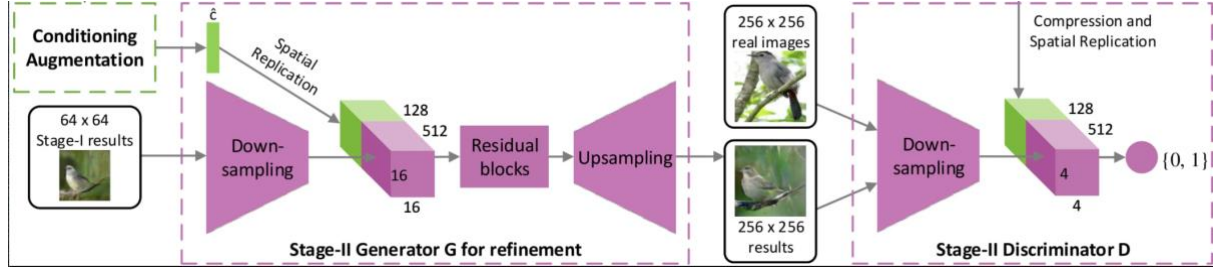


Figure 2.2.1 Stage 2 StackGAN [7]

## 2.4. Outcome of Literature Review

The plan is to implement a primitive DC-GAN & a Stack GAN of 2 stages and get a better loss function which prevents vanishing gradients and X Flat regions especially BCE loss in the training process and preferably get a human factor for future error analysis the, the end goal is to produce High-Res Images of textual descriptions, compare them based on their FID score and the Inception Score.

(More papers were referred, which are given in the References and citations sections)

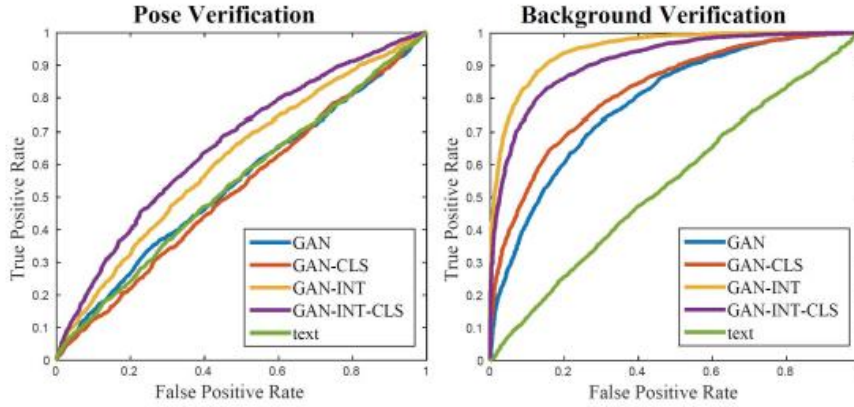


Figure 2.3.: ROC curves [3]

### 2.4.1. Drawbacks of DC-GAN

- It is very difficult to train GAN to generate high-resolution photo-realistic images from text descriptions.
- Simply adding more upsampling layers in state-of-the-art GAN models for generating high-resolution (e.g., 256×256) images generally results in training instability and produces nonsensical outputs.



Figure 2.4.1 Nonsensical outputs generated by DC-GAN

- The main difficulty for generating high-resolution images by GANs is that supports of natural image distribution and implied model distribution may not overlap in high dimensional pixel space.

## 2.5. Problem Statement

Stacked Generative Adversarial Networks (StackGAN) with Conditioning Augmentation for synthesizing photo-realistic images which will be the focus of this project along with its comparison to the DC-GAN put to the same task and how they matchup against each other and get statistical results.

## 2.6. Research Objectives

The proposed method decomposes the text-to-image synthesis to a novel sketch-refinement process. Stage-I GAN sketches the object following basic color and shape constraints from given text descriptions. Stage-II GAN corrects the defects in Stage-I results and adds more details, yielding higher resolution images with better image quality. Extensive quantitative and qualitative results demonstrate the effectiveness of our proposed method. Compared to existing text-to-image generative models, our method generates higher resolution images (e.g.,  $256 \times 256$ ) with more photo-realistic details and diversity.

Moreover establish a proper performance and error metrics and work towards optimizing the performance of the process and reduce the overall loss in the process , Using FID scores and the Inception Score to as a parameter to compare between the trained GANs.

# 3. Methodology and Framework

## 3.1. System Architecture

### 3.1.1. DC-GAN

The underlying idea is to augment the generator and discriminator in a GAN with suitable text encoding of the description. Conceptually, this is similar to conditioning the operation of the generator and discriminators on the text descriptions. The original work describes the implementation using Deep Convolutional Neural Networks hence the name DCGAN. The generator is a deconvolution network which

generates an image from the text based on noise distribution. The discriminator is a convolutional network which outputs the probability of the input image belonging to the original data distribution given the text encoding.

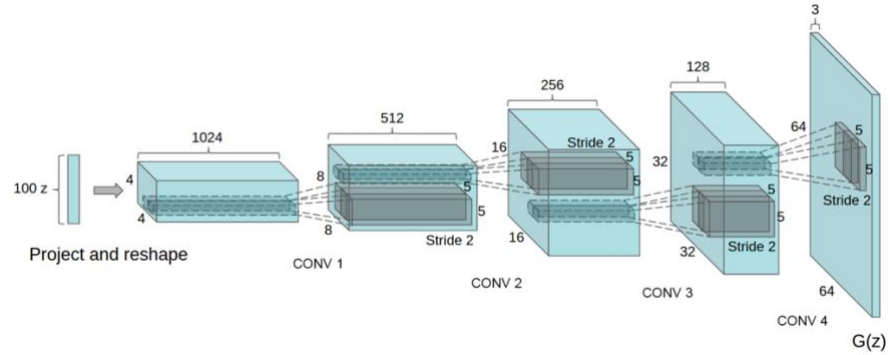


Figure 3.1 Basic DC-GAN Architecture [3]

It is the loss function of the network that we changed in to get varying results. As already mentioned the discriminator is a Convolutional Neural Network which was trained on two aspects:

1. It should be able to differentiate between a generated image and an original image for the same image description in text
2. The discriminator should be able to differentiate between a real image and fake text

### 3.1.2. StackGAN

To generate high-resolution images with photo-realistic details, we propose an easy yet effective Stacked Generative Adversarial Networks. It decomposes the text-to-image generative process into two stages

- Stage-I GAN: it sketches the primitive shape and basic colors of the thing conditioned on the given text description and draws the background layout from a random noise vector, yielding a low-resolution image.
- Stage-II GAN: it corrects defects within the low-resolution image from Stage-I and completes details of the thing by reading the text description again, producing a high-resolution photo-realistic image.



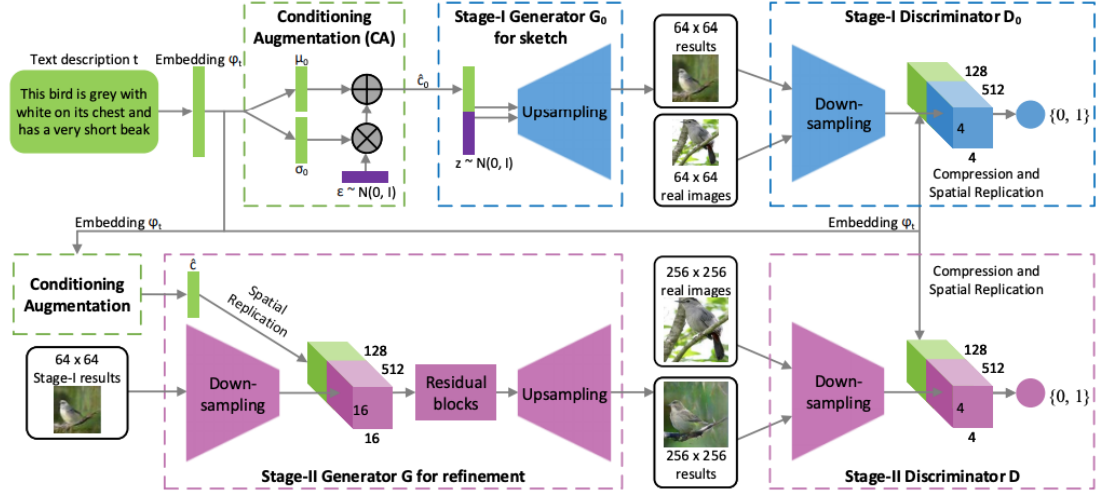


Figure 3.1 StackGAN Architecture [2]

## 3.2. Algorithms

Overall, the training procedure is similar to a two-player min-max game with the following objective function, [1]

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

Next, We have the Inception Score [12] which is calculated as :

$$\exp( KL(p(y|x)/p(y)) )$$

The score measures two things simultaneously:

- The images have variety
- Each image distinctly looks like something

If both things are true, the score will be high. If either or both are false, the score will be low.

To produce this score, we use a statistics formula called the Kullback-Leibler (KL) divergence . The KL divergence is a measure of how similar/different two probability distributions are.

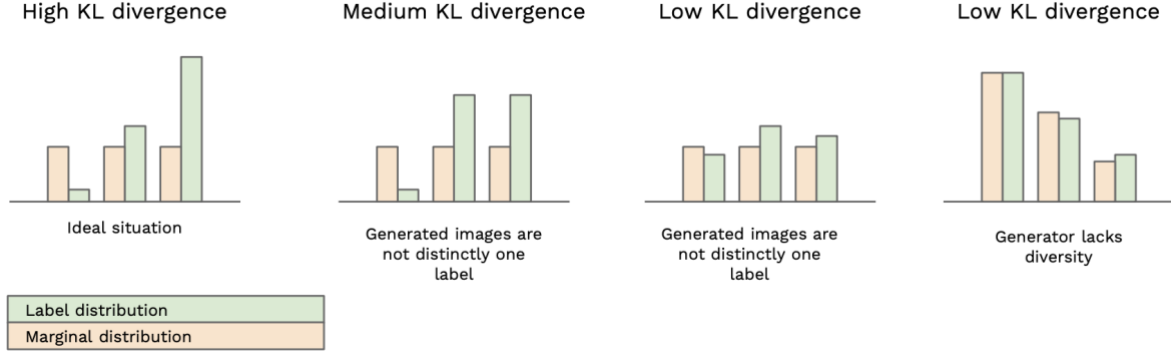


Figure 3.3.3 Implementation on the flowers Dataset [14]

KL divergence which is calculated as:

$$KL(\text{divergence}) = p(y|x) * (\log(p(y|x)) - \log(p(y)))$$

As, Inception Score has a lot of limitations, FID score [13] is implemented which is now the standard for measuring the performances of GANs today. In FID, we use the Inception network to extract features from an intermediate layer. Then we model the data distribution for these features using a multivariate Gaussian distribution with mean  $\mu$  and covariance  $\Sigma$ . The FID between the real images  $x$  and generated images  $g$  is computed as:

$$FID = |\mu - \mu_w|^2 + \text{tr}(\Sigma + \Sigma_w - 2(\Sigma\Sigma_w)^{1/2}).$$

Lower FID values mean better image quality and diversity.

### 3.2.1. Stage 1 StackGAN

For the generator  $G_0$ , to obtain text conditioning variable  $\hat{c}_0$ , the text embedding  $\phi_t$  (pre-trained) [16] is first fed into a fully connected layer to generate  $\mu_0$  and  $\sigma_0$  ( $\sigma_0$  are the values in the diagonal of  $\Sigma_0$ ) for the Gaussian distribution  $N(\mu_0(\phi_t), \Sigma_0(\phi_t))$ .  $\hat{c}_0$  are then sampled from the Gaussian distribution. Our  $N_g$  dimensional conditioning vector  $\hat{c}_0$  is computed by  $\hat{c}_0 = \mu_0 + \sigma_0$  is the element-wise multiplication,  $\sim N(0, I)$ . Then,  $\hat{c}_0$  is concatenated with a  $N_z$  dimensional noise vector to generate a  $W_0 \times H_0$  image by a series of up-sampling blocks. For the discriminator  $D_0$ , the text embedding  $\phi_t$  is first compressed to  $N_d$  dimensions using a fully-connected layer and then spatially replicated to form a  $M_d \times M_d \times N_d$  tensor. Meanwhile, the image is fed through a series of down-sampling blocks until it has  $M_d \times M_d$  spatial dimension. Then, the image filter map is concatenated along the channel dimension with the text tensor. The resulting tensor is further fed to a  $1 \times 1$  convolutional layer to jointly learn features across the image and the text. Finally, a fully connected layer with one node is used to produce the decision score.

### 3.2.2. Stage 2 StackGAN

We design Stage-II generator as an encoder-decoder network with residual blocks almost like the previous stage, the text embedding  $\phi_t$  is employed to generate the  $N_g$  dimensional text conditioning vector  $\hat{c}$ , which is spatially replicated to make a  $M_g \times M_g \times N_g$  tensor.

Meanwhile, the Stage-I result  $s_0$  generated by Stage-I GAN is fed into several down-sampling blocks (i.e., encoder) until it is a spatial size of  $M_g \times M_g$ . The image features and the text features are concatenated along the channel dimension. The encoded image features including text features are fed into several residual blocks, which are designed to find out multi-modal representations across image and text features. Finally, a series of up-sampling layers (i.e., decoder) are wont to generate a  $W \times H$  high-resolution

image. Such a generator is in a position to assist rectify defects within the input image while add more details to get the realistic high-resolution image. For the discriminator, its structure is analogous to that of Stage-I discriminator with only extra down-sampling blocks since the image size is larger during this stage. To explicitly enforce GAN to find out better alignment between the image and the conditioning text, instead of using the vanilla discriminator, we adopt the matching-aware discriminator proposed by Reed et al. for both stages. During training, the discriminator takes real images and their corresponding text descriptions as positive sample pairs, whereas negative sample pairs contains two groups. the primary is real images with mismatched text embeddings, while the second is synthetic images with their corresponding text embeddings.

### 3.3. Design Methodologies

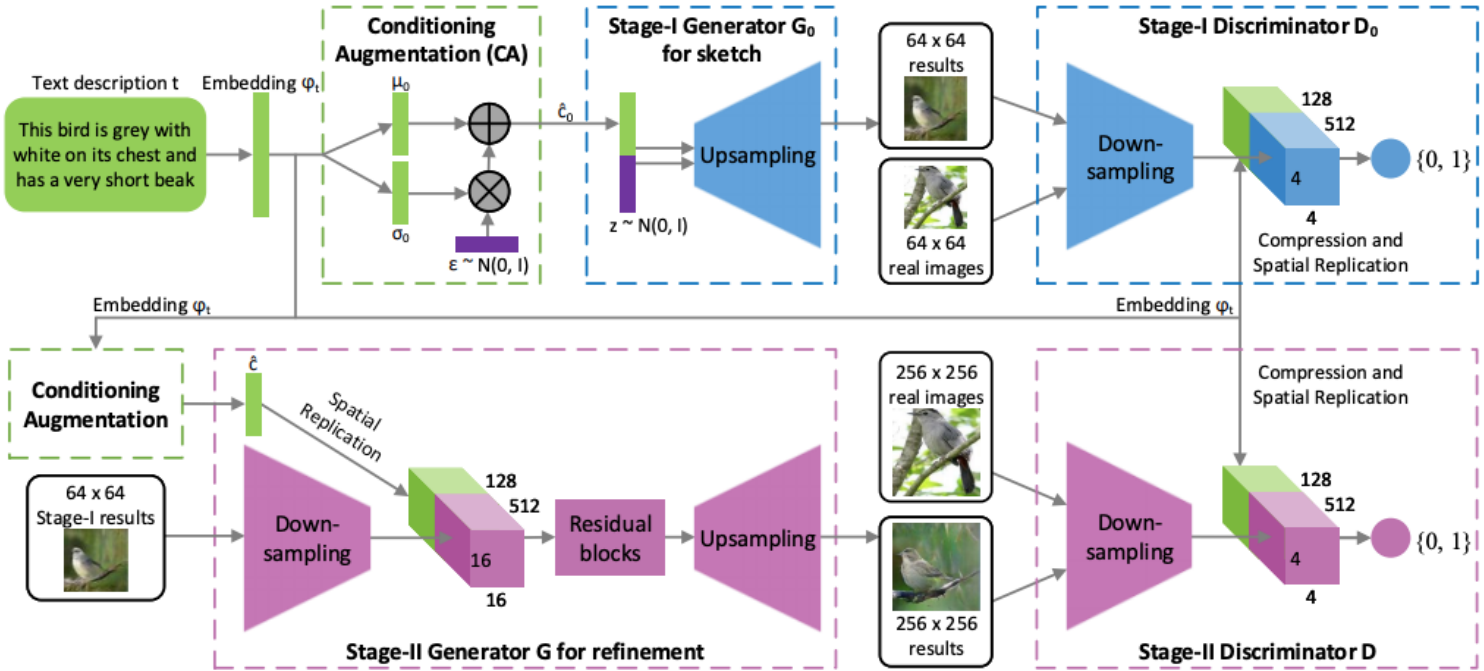


Figure 3.3.1 StackGAN working architecture [2]

Text description	This bird is blue with white and has a very short beak	This bird has wings that are brown and has a yellow belly	A white bird with a black crown and yellow beak	This bird is white, black, and brown in color, with a brown beak	The bird has small beak, with reddish brown crown and gray belly	This is a small, black bird with a white breast and white on the wingbars.	This bird is white black and yellow in color, with a short black beak
Stage-I images							
Stage-II images							

Figure 3.3.2 Final some examples on the birds Dataset [2]




























This flower is yellow in color, with petals that are vertically layered							
Stage-I images							
Stage-II images							
This flower has white petals with a yellow tip and a yellow pistil							
Stage-I images							
Stage-II images							
A flower with small pink petals and a massive central orange and black stamen cluster							
Stage-I images							
Stage-II images							

Figure 3.3.3 Implementation on the Oxford 102 Dataset [2]





Figure 3.3.4 Implementation on MS COCO Dataset [71]

## 4. Work Done

### 4.1. Images Generated by the different networks

#### 4.1.1. DC-GAN

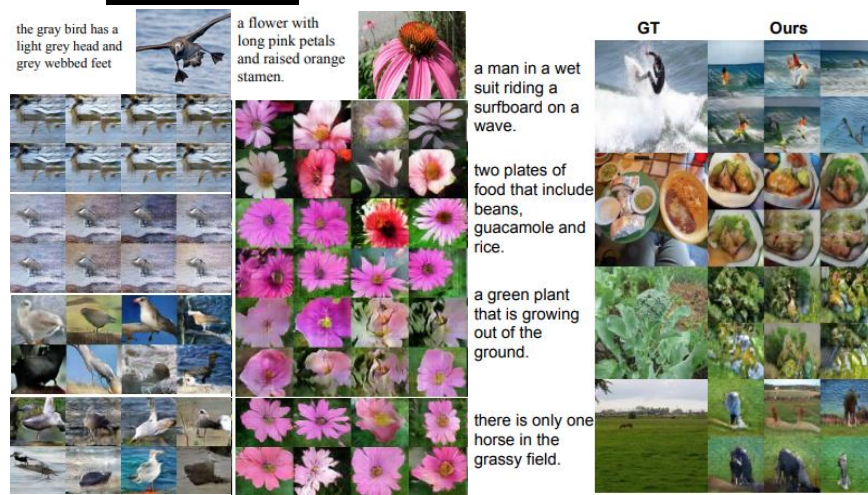


Figure 4.1.1: DC-GAN generated images(64x64) trained on the CUB dataset[8], Oxford 102 [9] dataset & MSCOCO [10]

### 4.1.2. StackGAN

This bird sits close to the ground with his short yellow tarsus and feet; his bill is long and is also yellow and his color is mostly white with a black crown and primary feathers



A large bird has large thighs and large wings that have white wingbars



This smaller brown bird has white stripes on the coverts, wingbars and secondaries



A cardinal looking bird, but fatter with gray wings, an orange head, and black eyerings



Figure 4.1.2: Collection of StackGAN generated images (stage wise) trained on the CUB [8] dataset along with their text input embed

## 4.2. Summary

- Sorted the MOC ,Flowers and the MC-COCO datasets
- Pre-processed the data.
- Trained a DC-GAN for this task [1] [3].
- Trained both the stages of the StackGAN for the same.
- Completed the evaluation phase.
- Implemented the Inception Score and FID scores on the trained GANs.
- Compiled the final statistical results.
- Created a smaller model for demonstration implementing and keeping the original model but on a smaller scale to ease the demonstration purpose to get a good view of all the GAN model work on the given task of Text2Image generation.

### 4.3. Results & Final Scores

In GANs, the objective function for the generator and the discriminator usually measures how well they are doing relative to the opponent. For example, we measure how well the generator is fooling the discriminator. It is not a good metric in measuring the image quality or its diversity.

So, We use Inception Score and Fréchet Inception Distance to measure their performances.

See how they both work ([Pg. 9](#))

Results and scores for the StackGAN [\[3\]](#):

METHOD	CA	TEXT TWICE	INCEPTION SCORE
<b>64×64 STAGE-1@ GAN</b>	no	/	2.66 ± .03
	yes	/	2.95 ± .02
<b>256×256 STAGE-1@ GAN</b>	no	/	2.48 ± .00
<b>256×256 STAGE-1@ GAN</b>	yes	/	3.02 ± .01
<b>128×128 STACKGAN</b>	yes	no	3.13 ± .03
<b>128×128 STACKGAN</b>	no	yes	3.20 ± .03
<b>128×128 STACKGAN</b>	yes	yes	3.35 ± .02
<b>256×256 STACKGAN</b>	yes	no	3.45 ± .02
<b>256×256 STACKGAN</b>	no	yes	3.31 ± .03
<b>256×256 STACKGAN</b>	yes	yes	3.70 ± .04

TABLE 2: Inception scores calculated with 30,000 samples generated by different baseline models of our StackGAN [\[3\]](#)

\*Metrics used and combined and differentiated by different papers referenced below:

- GAN-INT-CLS (Original DC- GAN ) [\[3\]](#)
- GAWWN (Generative Adversarial What-Where Network (GAWWN) [\[15\]](#)
- StackGAN (Original StackGAN) [\[2\]](#)

Metric	CUB			Oxford		COCO	
Model Used	GAN-INT-CLS	GAWWN	StackGAN	GAN-INT-CLS	StackGAN	GAN-INT-CLS	StackGAN
→							
FID ↓	68.79	67.22	51.89	79.55	55.28	60.62	74.05
FID# ↓	68.79	53.51	35.11	79.55	43.02	60.62	33.88
IS ↑	2.88 ± .04	3.62 ± .07	3.70 ± .04	2.66 ± .03	3.20 ± .01	7.88 ± .07	8.45 ± .03



$IS^\# \uparrow$	$2.88 \pm .04$	$3.10 \pm .03$	$3.02 \pm .03$	$2.66 \pm .03$	$2.73 \pm .03$	$7.88 \pm .07$	$8.35 \pm .11$
$HR \downarrow$	$2.76 \pm .01$	$1.95 \pm .02$	$1.29 \pm .02$	$1.84 \pm .02$	$1.16 \pm .02$	$1.82 \pm .03$	$1.18 \pm .03$

TABLE 3: Inception scores (IS), Fréchet inception distance (FID) and average human ranks (HR) of GAN-INT-CLS [35], GAWWN [33] and our StackGAN-v1 on CUB, Oxford-102, and COCO. (# means that images are re-sized to 64×64 before computing IS\* and FID\*)

Note : Inception Score the higher the better and lower the FID the better (as marked by the arrows or for more details [\(Pg. 9\)](#))

## 5. Conclusion & Future

- After the analysing the scores (FID & Inception scores) we can see how the StackGAN improves the GAN architecture and increases the resolution and improves the image quality by a lot as we can see by the trained models' images generated.
- Compared to existing text-to-image generative models, StackGAN generates higher resolution images (e.g., 256×256) with more photo-realistic details and diversity.
- The double stage GAN proves to be a crucial part in improving the scores by a huge margin. The double structure consisting of 2 generators and 2 discriminators upsamples the image to a higher resolution (256x256) instead of the low-res(64x64) image generated by the DC-GAN.
- StackGAN is a powerful concept and the fact that they are able to create such photo-realistic images from the text is simply incredible.
- Moreover, training and implementing other GANs like the OP-GAN, AttnGAN, BigGAN or combing StackGAN with other models etc for the current objective of generating photo-realistic images from textual descriptions (combing GANs with CLIP).
- In addition to this, Due to Quarantine the resources that were used on this project were limited (graphical memory), The training time could have been reduced by a huge margin with the appropriate graphical memory.
- The full code on the implementation of the GANs and the presentation is available at : [Github repository](#)



## 6. References

- [1] Zhu, M., Pan, P., Chen, W., & Yang, Y. (2019). Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5802-5810).
- [2] Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., & Metaxas, D. N. (2017). 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).
- [3] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., & Lee, H. (2016, June). Generative adversarial text to image synthesis. In *International Conference on Machine Learning* (pp. 1060-1069). PMLR.
- [4] Gao, L., Chen, D., Zhao, Z., Shao, J., & Shen, H. T. (2021). Lightweight dynamic conditional GAN with pyramid attention for text-to-image synthesis. *Pattern Recognition*, 110, 107384
- [5] Sun, J., & Zhang, B. (2019, December). MCA-GAN: Text-to-Image Generation Adversarial Network Based on Multi-Channel Attention. In *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)* (Vol. 1, pp. 1845-1849). IEEE.
- [6] Arjovsky, M., Chintala, S., & Bottou, L. (2017, July). Wasserstein generative adversarial networks. In *International conference on machine learning* (pp. 214-223). PMLR.
- [7] Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., & Metaxas, D. N. (2018). Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(8), 1947-1962.
- [8] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. *The Caltech-UCSD Birds-200-2011 Dataset*. Technical Report CNS-TR2011-001, California Institute of Technology, 2011.
- [9] M.-E. Nilsback and A. Zisserman. *Automated flower classification over a large number of classes*. In ICCVGIP, 2008.
- [10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollr, and C. L. Zitnick. *Microsoft coco: Common objects in context*. In ECCV, 2014
- [11] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. FeiFei. *ImageNet Large Scale Visual Recognition Challenge*. International Journal of Computer Vision (IJCV), 115(3):211–252, 2015.
- [12] Salimans, Tim, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. "Improved techniques for training gans." *arXiv preprint arXiv:1606.03498* (2016).
- [13] Heusel, Martin, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. "Gans trained by a two time-scale update rule converge to a local nash equilibrium." *arXiv preprint arXiv:1706.08500* (2017).
- [14] Goldberger, Jacob, Shiri Gordon, and Hayit Greenspan. "An Efficient Image Similarity Measure Based on Approximations of KL-Divergence Between Two Gaussian Mixtures." In *ICCV*, vol. 3, pp. 487-493. 2003.
- [15] Reed, Scott, Zeynep Akata, Santosh Mohan, Samuel Tenka, Bernt Schiele, and Honglak Lee. "Learning what and where to draw." *arXiv preprint arXiv:1610.02454* (2016).
- [16] S. Reed, Z. Akata, B. Schiele, and H. Lee. Learning deep representations of fine-grained visual descriptions. In CVPR, 2016.