iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 Issue: IV Month of publication: April 2018

DOI: http://doi.org/10.22214/ijraset.2018.4698

Comparison of Text to Image Synthesis Algorithms.

Prachi Patil[1] , Pranav Dange[2] , Akshay Patade[3] , Shubham Pawar[4]
[1]Assistant Professor Information Technology at Fr CRCE, Mumbai University
[2]Final year Information Technology student at Fr CRCE, Mumbai University,
[3]Final year Information Technology student at Fr CRCE, Mumbai University,
[4]Final year Information Technology student at Fr CRCE, Mumbai University,

Abstract: As in the recent times there has been significant progress in the generation of photo realistic images based on text descriptions. There have been various models achieving great results in this domain which have produced remarkable results. This paper aim's to analyze two such successful models and to analyze their efficiency and to compare the result in generating photo realistic images based on the text input. The purpose is also to look into the architecture and also check how efficiently the results are produced and also analyze the pros and cons of the two models. Further it would also look into the future aspects of these models so as to how these models can be used as a base to build upon more complex architecture.

Keywords: StackGan, StackGan++, conditional augmentation

I. INTRODUCTION

The generation of new image from manipulating given set of related images is a time consuming and tedious tasks for human but the advances in the deep learning domain has made artificial generation of images relatively easy. There are many Generative Adversarial Networks present in doing so. The first method was proposed by Ian Goodfellow which had a basic generator and discriminator model [3].Improvement on this approach was presented by StackGan-v1 which had a two-stage model for generating high resolution photo-realistic images[1].A further better model is StackGan++ which has higher stability and is an improvement on all the previous models[3][2][1]. There are many approaches in generating photo-realistic images but we are going to see two most successful approaches which are able to successfully generate photo-realistic images [1][2]. The aim of this paper is successfully study these two approaches see the merits and demerits in each case and compare theirs outputs.

II. GENERATIVE ADVERSARIAL APPROACHES FOR GENERATION OF PHOTO REALISTIC IMAGES

A. Data Source

1) *Environmental Setup:* Setup: In order to undertake the experiments and evaluate the results from the experiments, Oxford-102[15] and CUB database was chosen. This contains 8,189 images of flowers from 102 different categories. The training and testing data is divided in the ratio of 85:15 to check the efficiency of the models. Using this database we generate the output from both this model and see the accuracy and clarity of generation.

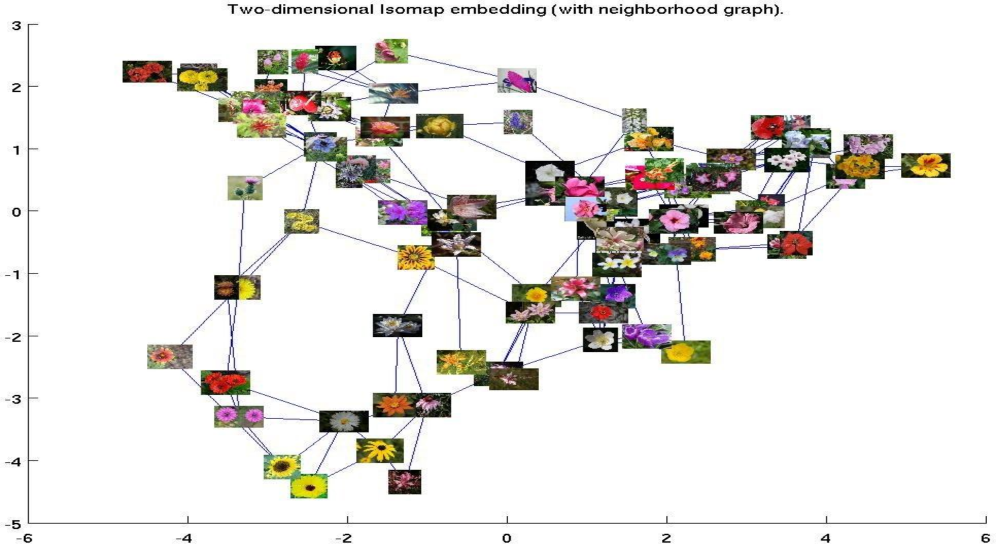


Fig 1: The above figure shows stage I two-dimensional isomap embedding (with neighborhood graph).

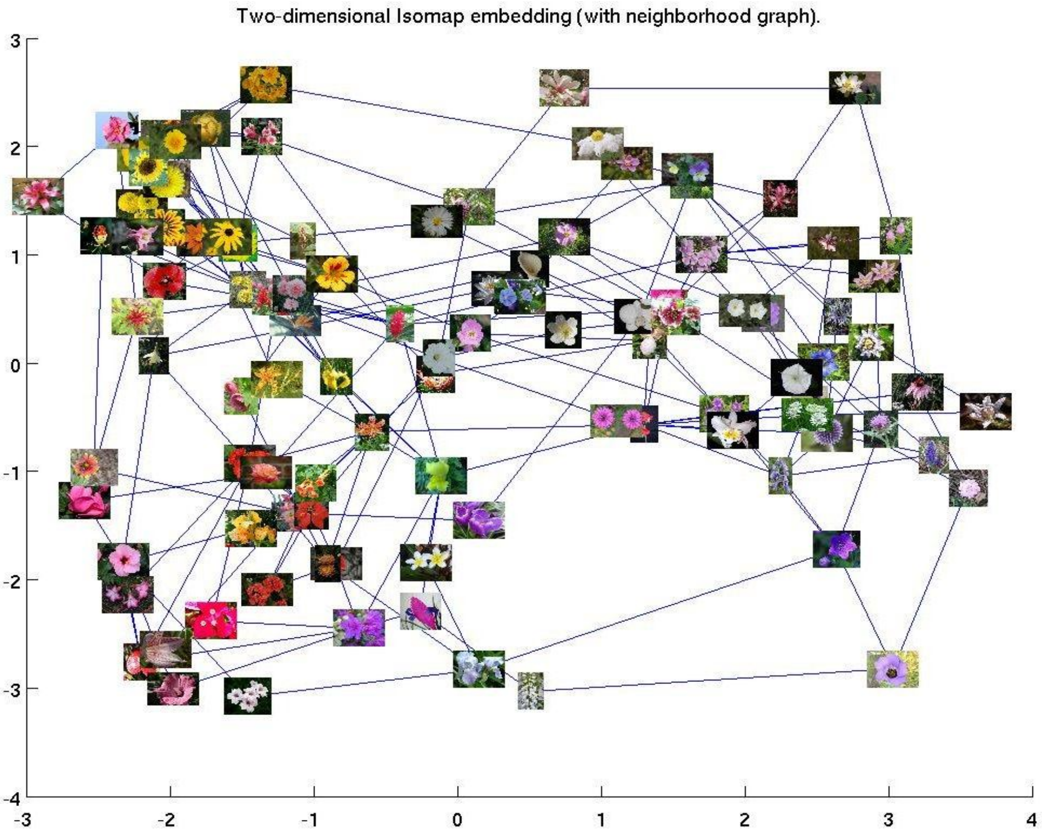


Fig 2: The above figure shows stage II two-dimensional isomap embedding (with neighborhood graph).

B. Overview of the Models used

*1) StackGAN-v1:*StackGAN is the improved version of GAN algorithm. In Earlier GAN algorithms the generator and the discriminator used to work simultaneously in order to generate the photo-realistic images. But the images which were generated by the GAN algorithm were of the resolution 64*64 pixels which were not clear. That is why StackGAN-v1 was introduced. In StackGan-v1 the Stage-1 generates a low resolution image(64x64) with primitive shape and basic required colour based on the text embedding produced by conditioning augmentation process[10][11].Then this low resolution image is passed to the Stage-2 which corrects the image and also adds additional details missed by the previous stage to generate a high quality image(256x256).

As there is very less text available for learning we make use of conditioning augmentation for getting better result over a limited amount of text. So the text description t is first encoded

by an encoder, yielding a text embedding φt which is further used to calculate the mean μ(φ t) and diagonal covariance matrix Σ(φ t) then it is randomly sample the latent variables ĉ

from an independent Gaussian distribution N (μ(φ t), Σ(φ t)) .This Conditioning Augmentation gives more training pairs given a small number of image-text pairs, and this thus useful even with limited amount of data. For smoothness and overfitting, it uses Kullback-Leibler divergence (KL divergence) between the standard Gaussian distribution and the Gaussian distribution condition [12], [13].

2) *Equation for StackGAN:* $\min_{G} \max_{D} V (D, G) = E_{x\sim p\, data} [\log D(x)] + E_{z\sim p\, z} [\log(1 - D(G(z)))]$(1)

3) *Equation for Conditional Augmentation:*

$$D_{KL} (N (\mu(\varphi_t), \Sigma(\varphi_t)) \,\|\, N (0, I))$$

C. Result

1) *Text Description:* -Flower which is purple red and white in color and which has multi colored petals
2) *Stage I result*

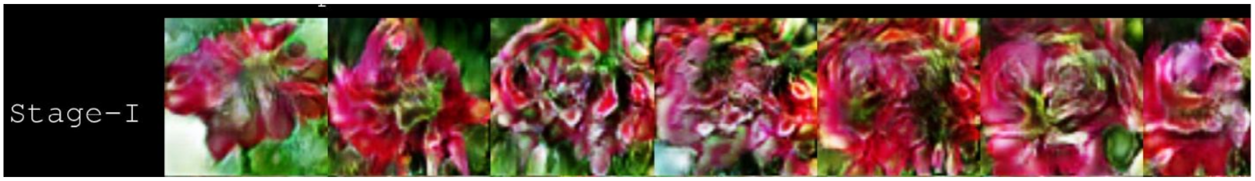


Fig 3: The above figure shows Stage I result

3) *Stage II result*

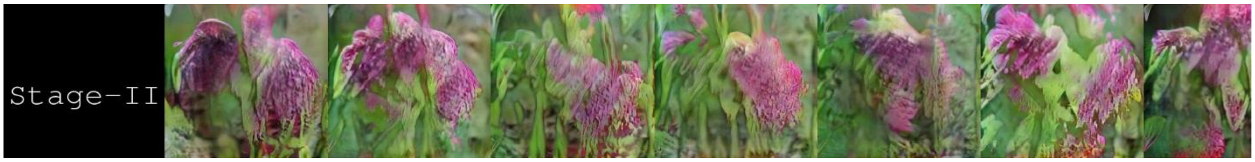


Fig 4: The above figure shows Stage II result.

4) *StackGAN-v2:* : [1][2] This model uses advanced multi-stage generative adversarial network architecture. It is created for both conditional and unconditional generative tasks. This StackGAN-v2 comprises of multiple generators (Gs) and discriminators (Ds) in a tree-like structure. Images of different scales corresponding to the same scenario are generated from different branches of the tree. At every branch, the generator collects the image distribution at that scale and the discriminator calculates the probability that a sample came from training images of that scale rather than the generator. The generators are build to approximate the multiple distributors. The generators and distributors are trained in an alternating fashion. The Stack-GAN-v2 behaves more stable than StackGAN-v1 by approximating many distributors. There are two types of multi-distributions
5) *Multi-scale image distributions approximation:* The StackGAN-v2 framework has a tree-like structure. It takes a noise vector as the input and has many generators to generate images of direct scales. In the multi-scale image distributions approximation, the multiple image distributions increase the chance of data distributions sharing supports with model distribution. At the first branch low-resolution image distribution results in images with basic structures and colour. Then the generators at the below branches can generate high resolution images.
6) *joint Conditional and Unconditional Image distributions Approximation:* In conditional image generator, the generator determines whether the image and the condition match or not. In unconditional image generator, the loss determines whether the image is real or fake.

D. Result

1) *Text description:* - Flower which has long thin yellow petals and has a lot of yellow anthers in the center
2) *Stage I result:*

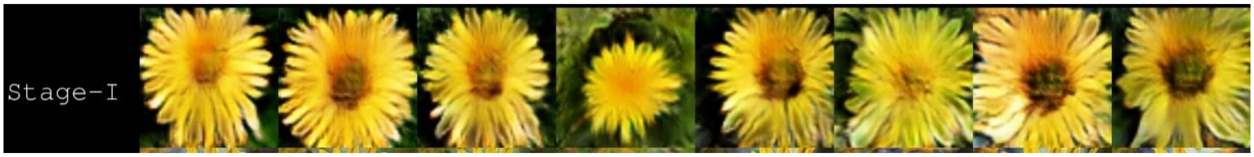


Fig 5: The above image shows Stage I result

3) *Stage II result:*

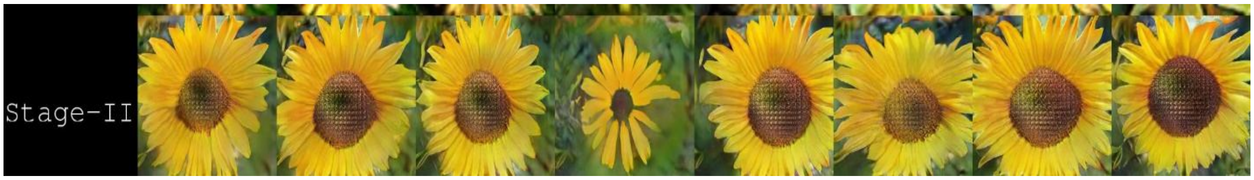


Fig 6 : The above figure shows Stage II result.

E. Advantages

As compared to the previous version this generates a comparatively better resolution image.

It captures more detail due to separation in stages for generation.

F. Drawbacks

Can keep on generating certain pattern of images which would lead to wrong output.

G. Comparison between StackGAN-v1 and StackGAN-v2

From the above two models discussed it is been seen that StackGAN-v2 generates more realistic images than StackGAN-v1 .The main problem of StackGAN-v1 is that if the Stage-1 image is not related to the text the Stage-2 output will completely be differ from the desired result. That is why in StackGAN-v2 multiple generators are used so that each generator has a different image and all of them don't produce same results or same pattern of images as in the case of StackGan-v1. Also presence of multiple generators make the model more stable but the only disadvantage is that it makes the model more slow and requires high hardware support[1][2]. Also from the results it is been seen that StackGAN-v1 can't handle complex sentences or the more detailed text description. This problem is overcomed by StackGAN-v2.

III. RESULTS & CONCLUSION

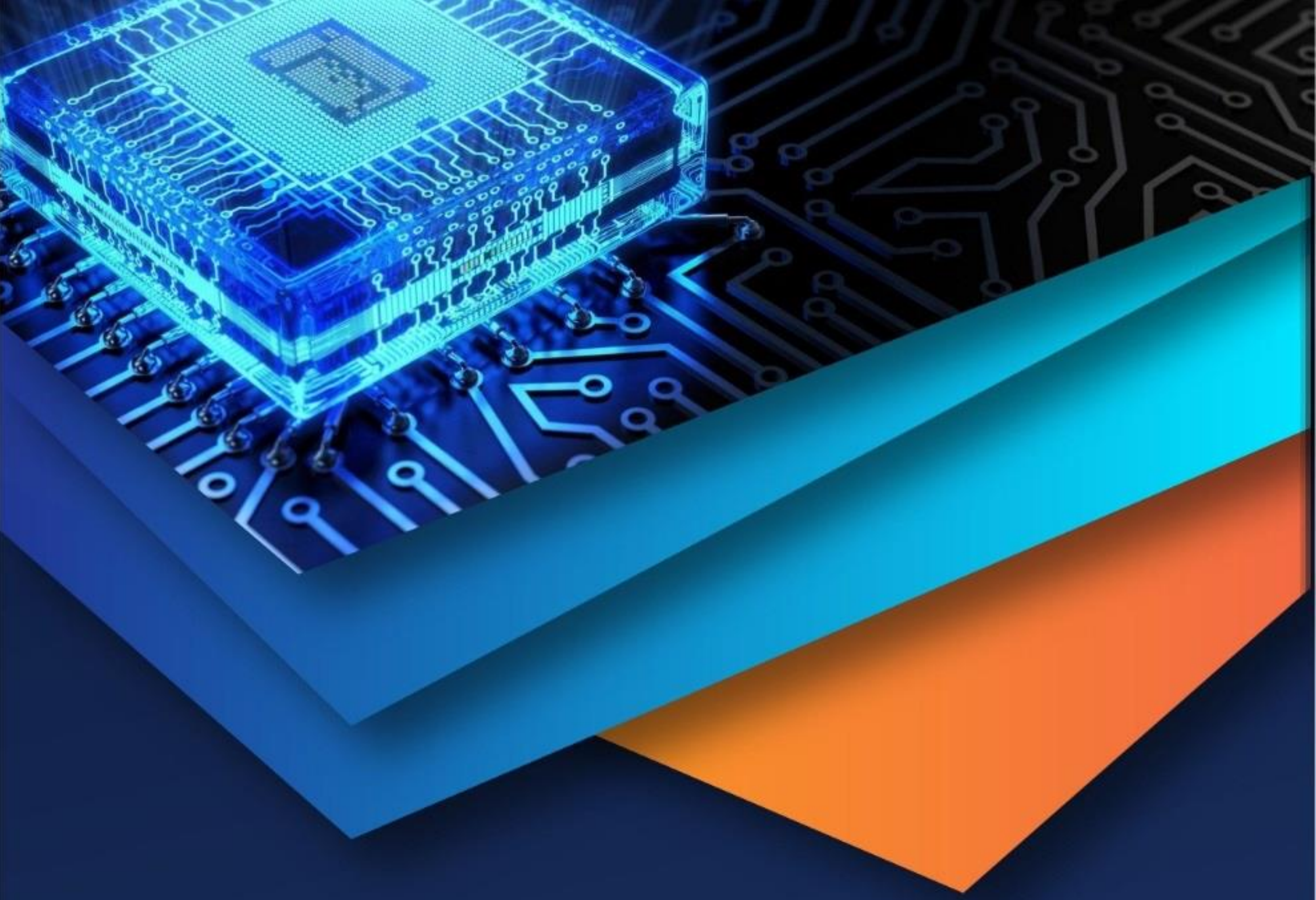
Table I: Inception score of StackGAN-v1 and StackGAN-v2

Metric	Dataset	StackGAN-v1	StackGAN-v2
Inception Score	CUB	3.70±.04	4.04±.05
	Oxford	3.20±.01	/

The results from Table 1 shows that StackGAN-v2 algorithm gives better inception score than Stackgan-v1 algorithm for CUB dataset. In this paper we illustrated the comparison of various GAN algorithms for image synthesis. This paper compared the models using the inception score. Thus, the comparative analysis concludes that StackGAN-v2 is a better algorithm for image synthesis because it has got maximum inception score of around 4.04±.05.

REFERENCES

[1] Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Network
[2] StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Network
[3] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley,S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets.In NIPS, 2014
[4] M. Arjovsky and L. Bottou. Towards principled methods for training generative adversarial networks. In ICLR, 2017
[5] C. K. Snderby, J. Caballero, L. Theis, W. Shi, and F. Huszar. Amortised map inference for image super-resolution. In ICLR, 201
[6] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In ICLR, 2016
[7] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training GANs. In NIPS, 2016
[8] M. Arjovsky, S. Chintala, and L. Bottou.Wasserstein GAN. arXiv:1701.07875, 2017
[9] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li. Mode regularized generative adversarial networks. In ICLR, 201
[10] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. In ICLR, 201
[11] S. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee. Learning what and where to draw. In NIPS, 201
[12] C. Doersch. Tutorial on variational autoencoders. arXiv:1606.05908, 201
[13] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. In ICML, 2016
[14] S. Reed, Z. Akata, B. Schiele, and H. Lee. Learning deep representations of fine-grained visual descriptions. In CVPR, 2016
[15] http://www.vision.caltech.edu/visipedia/CUB 200-2011.html, “Bird database.”



10.22214/IJRASET

INDEX COPERNICUS
45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL
IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY