3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE-2020), 07-08 February 2020, (IEEE Conference Record # 48199)

Text to Image Synthesis using Residual GAN

Dr. Priyanka Mishra
Computer Science & Engineering
Indian Institute of Information Technology, Kota, India
priyanka.cse@iiitkota.ac.in

Tribhuvan Singh Rathore
Computer Science & Engineering
Indian Institute of Information Technology, Kota, India
2016kucp1044@iiitkota.ac.in

Shivani
Computer Science & Engineering
Indian Institute of Information Technology, Kota, India
2016kucp1049@iitkota.ac.in

Sachin Tendulkar
Computer Science & Engineering
Indian Institute of Information Technology, Kota, India
2016kucp1047@iiitkota.ac.in

Abstract— **In the world of computer vision, a very intriguing problem is synthesizing or generating images (from the noise) of the reasonable quality from text descriptions. The applications of this problem are immense such as photo-editing, computer-aided design, etc. But the current AI systems are not up to the mark to reach the desired outcome. However, in recent years the progress in the field of text classification and image classification fields have paved the way for more advanced AI systems that can be used to achieve the desired goal by utilizing the discriminative power and strong generalization properties of attribute representations of recurrent neural networks and convolutional neural networks. Meanwhile, GANs have proved to produce reasonable images of birds, flowers, etc. In this work, we present GAN architecture to effectively aid the translation visual concepts from the text to image.**

Index Terms: GAN, Generator, Discriminator, Text to Image, Residual GAN.

I. INTRODUCTION

One of the most complex and interesting problems in Natural Language Processing and Computer Vision is image captioning: provided with an image, the system produces a text description of the image. Text to image synthesis is the opposite of the above problem: provided with a text description, the system needs to generate an image having the visual properties as described in the description. Generating images of vivid details and clarity from normal English text queries describing the visual quality of images is one of the superior utilizations of the novel conditional generative models. In this work, our objective is to create an image of acceptable quality from the text input.

From an abstract point of view, this problem is quite similar to the language translation problem. Just like, alike interpretations can be encoded in two different languages. Here, images and text can be considered as two different types of languages to encode alike interpretations. This type of complete observed knowledge about an object can be maintained in attribute representations distinguishing characteristics of the object category encoded into a vector which requires the discriminative power and strong generalization properties of attribute representations. But this requirement poses a problem as text to image or image to text conversions are highly multi modal in the sense that there are many acceptable structures of one based on another i.e. the texts that correctly describes the image and vice versa. But this difficulty can be resolved by the point that language is often sequential in nature so a given text query can be disintegrated sequentially i.e. predicting the next word conditioned on all previous words and the image.

Our motivation for this work was based on the recent progress of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) have begun to produce profoundly discriminative and generalizable text descriptions determined automatically from characters and words. This task can be viewed as the combination of the following tasks, the first is to extract the important visual information from the text. In this task obtaining the context is of little help so we cannot use word2vec as Word2Vec is not really useful as the context of the word is unable to capture the visual properties as well as an embedding explicitly trained to do so. And then use these learned features to generate images. Here the main aim of our work is to develop a GAN architecture (Residual GAN or RGAN) that generate flower images of reasonable visual details from the given text query.

II. DATASETUSED

We have used a publicly available dataset Oxford-102 flowers dataset. This dataset comprises of 8,192 images of 102 categories of flowers. Each category of flower contributes from 40 and 258 images. This dataset includes only photos, but no descriptions. So, we have used publicly available captions collected by Reed[8]. For every image, there are ten descriptions corresponding to it. Each of the descriptions has a minimum length of 10 words and they don't define the background of the image and the flower's species.

III. BACKGROUND

Text to image synthesis is based on GAN and in this section, we briefly describe the about the working of GAN [6].

A. Generative Adversarial Network(GAN)

The principal thought following GAN [6] is to determine two networks- a Generator network G which attempts to produce images, and a Discriminator network D, which attempts to differentiate between 'real' and 'counterfeit' (generated) images. D learns to map features extracted from the images to the labels (real or fake) focused entirely on that correlation. On the other hand, G rather than predicting a label provided specific characteristics it strives to predict characteristics provided a specific label. GANs consist of G and D indulging with each other in a two-player minimax game: The discriminator attempts to differentiate real images from counterfeit images, while the generator attempts to deceive the discriminator.

978-1-7281-1683-9/20/$31.00 ©2020 IEEE

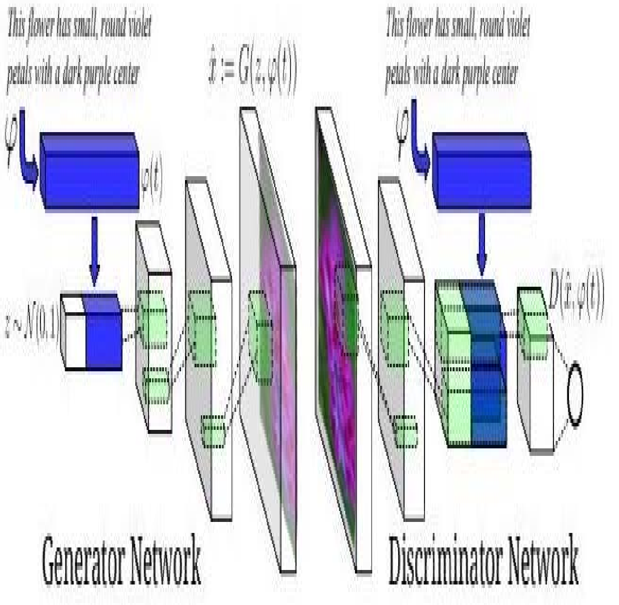

This flower has small, round violet petals with a dark purple center
φ
$\varphi(t)$
$z \sim \mathcal{N}(0,1)$
$\hat{x} := G(z, \varphi(t))$
This flower has small, round violet petals with a dark purple center
φ
$D(\hat{x}, \varphi(t))$
Generator Network
Discriminator Network

Fig. 1. Conditional GAN architecture. Text encoding $\phi(t)$ is used in both generator and discriminator [1]

$$minGmaxDV(D,tt) = Ex \sim p_{data(x)}[logD(x)] + Ex \sim p_{z(z)}[log(1 - D(tt(z))] \qquad (1)$$

Discriminator minimizes its loss when D(x) is equal to 1 and D(G(z)) is equal to 0, that is, when the discriminator's probability is 1 for real image and 0 for fake or synthesized image. While the generator attempts to maximize D(G(z)) meaning generator attempts to produce such type of images that discriminator thinks of as real images. It has proved profitable for the generator to maximize log(D(G(z))) instead of diminishing log (1- D(G(z))).–

GANs can be conditioned on different variables leading to the generated images conditioned on variables [1].

$$minwGmaxwDV(D,tt) = Ex \sim p_{x(x)}[logD(x,wD)] + Ex \sim p_{z(z)}[log(1 - D(tt(z,wG),wD)] \qquad (2)$$

Where z is the underlying "code" that is usually sampled from an uncomplicated distribution (such as normal distribution). Conditional GAN is a form of GAN where both generator and discriminator gets extra conditioning variables c, producing G (z, c) and D (x, c). This form of GAN enables G to produce images conditioned on variables c (here text).

IV. METHOD

One of the problems in our objective was how to represent text in such a format that it is visually discriminative. To solve this we take a CNN that will give the vector embedding of the image, and an RNN that utilizes LSTM at its core to convert text into vector form. After having text and image embeddings now we define a loss function that we will like to minimize using these embeddings such that the intuition behind it will be that "A text encoding should have a higher compatibility score with images of the corresponding class compared to any other class and vice-versa"[1].

x = correct text embedding, v = correct image embedding, x_w= incorrect text embedding, v_w= incorrect image embedding, ζ= cosine similarity

$$Loss = max(0, \alpha - \xi(x,v) + \xi(x,vw)) + max(0, \alpha - \xi(x,v) + \xi(xw,v)) \qquad (3)$$

The resulting gradients are back propagated through both CNN and RNN so that both learns similarity between correct text andcorrectimage.Wehavechosenalphatobe0.02.

V. RELATEDWORK

Generative adversarial networks established the adversarial learning procedure and described for the first time a basic GAN formulation and its training procedure [6]. It utilized convolutional decoder networks, for the generator network module and convolutional decoder networks for the discriminator.

After the creation of GAN in 2014, a lot of people have freshly utilized the ability of deep convolutional decoder networks to produce lifelike vivid images. Deconvolution network is trained (which consists of many conv2d and up sampling layers) to generate 3D chair renderings accustomed to a set of graphics regulations designating shape, posture, and brightness [11].

A lot of work has been done in improving the resolution of the generated images but those models were not conditioned on any other external variable as demonstrated in Ryan Dahl's work in Pixel Recursive Super Resolution. Producing high-resolution images with sharp and vivid details conditioned on an external variable is still an active area of research requiring a lot of work to be done to gain some fruitful results.

Author, utilized the notion of generation of images condition on external variable by introducing the fact that the model needs to be conditioned on text descriptions instead of class labels in order to capture visual information contained in the text [5].

The main qualification of our work from the previous conditional GANs conditioned on text works described above is that our model has certain architectural features designed to improve the overall efficiency of the model while producing acceptable results[9]. A similar work has been done on image restoration, we have also considered their work for the use of residual connection in generator and discriminator[10]. Due to the recent progress in recurrent neural network decoders, they have been used to generate text summaries accustomed on images [12][13].

VI. RGAN NETWORK ARCHITECTURE

Based upon the idea of using residual connection[10] we have created following architecture. We use the following notation. The generator network is denoted $tt{:}R{*}R^{T} \rightarrow R^{I}$ And the discriminator network is denoted as

$R * R^{T} \rightarrow \{0, 1\}$, where T is the dimension of the text description embedding, I is the dimension of the image, and Z is the dimension of the noise input . Generator takes random noise and text embedding as input and maps them to output image while discriminator takes an image and text embedding and maps them to either 0 in case of image is generated and 1 in case of image is real i.e not generated.

A. Generator

First, sample from the noise prior $zRzN$ (0, 1) and encode the text query t using rnn and these are given as input to the generator. Following steps are taken by the generator:

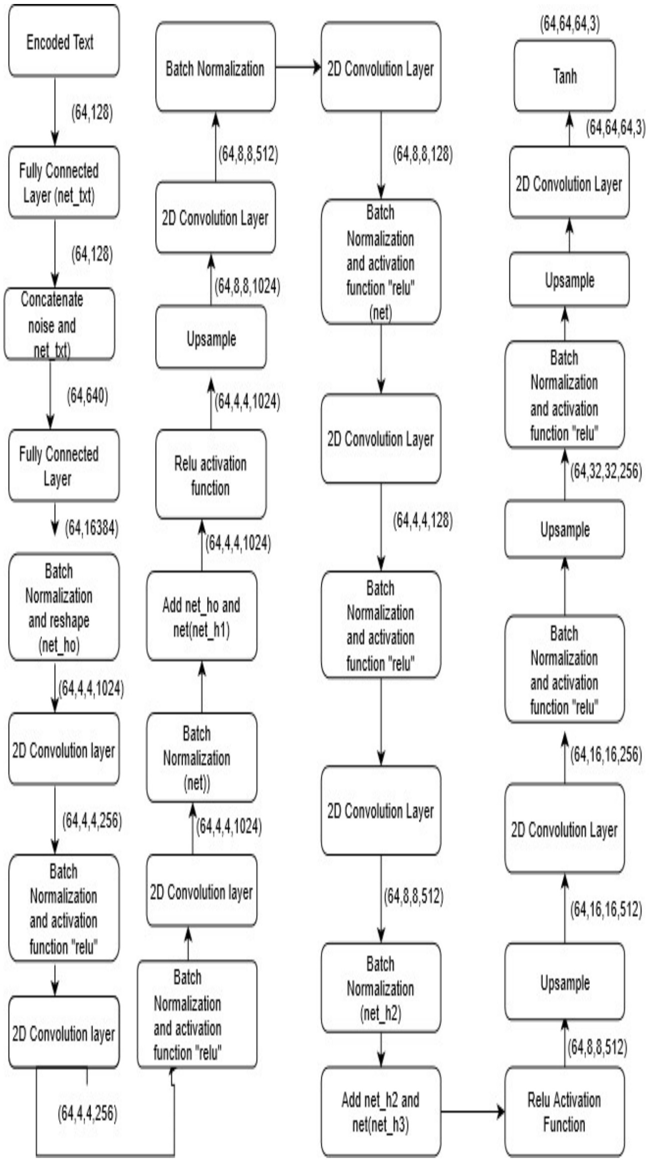


Fig. 2. RGAN Generator Architecture

B. Discriminator

The discriminator takes in an image and a text query and it outputs whether the image is generated or not. The following are the steps taken by discriminator:

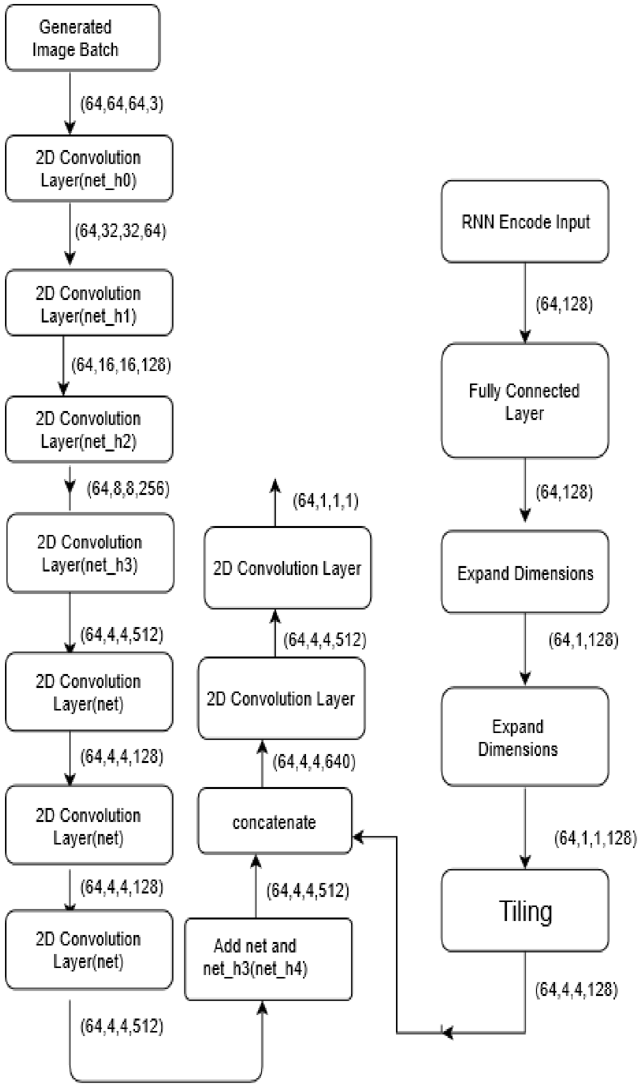


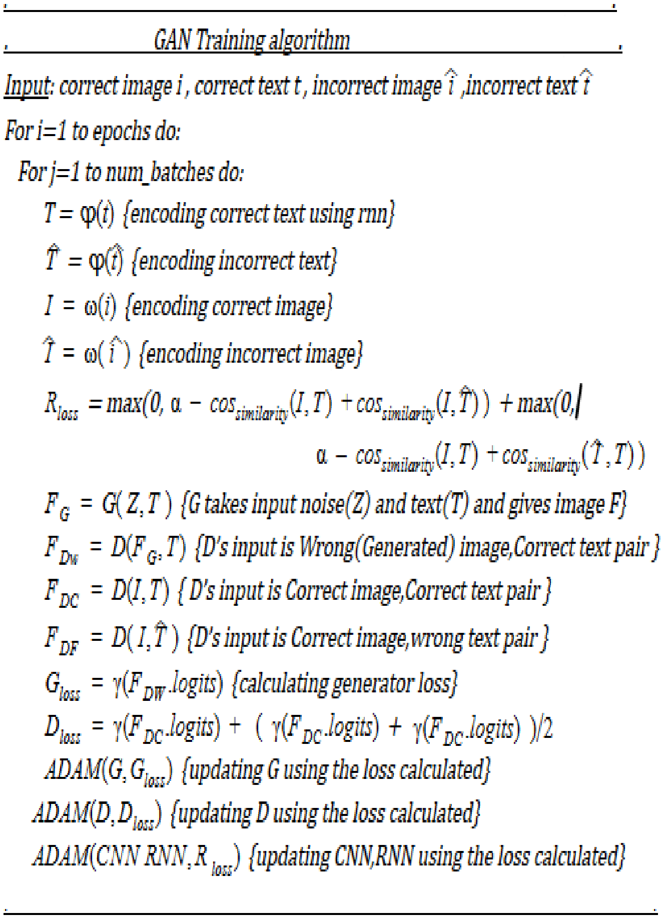
Fig. 3. RGAN Generator Architecture

VII. TRAINING ALGORITHM

We train both the generator and the discriminator to make them stronger together so that the ability of generator and discriminator is not too imbalanced because if the discriminator learns "too much", the gradients propagated by the discriminator will be very small and the generator will not learn anything.

While training the GAN (text, image) pair is required to be looked at as a single unit. During the first training iterations, the generator will generate noise so, in the beginning, the discriminator will learn to tell real images apart from that noise. Then the generator improves the quality of images and also learns to sync them with the visual information conditioned by the text and likewise discriminator will also learn to evaluate whether the images from the generator meet the condition or not.

The training regime of the discriminator is purely supervised for it to know that in a given text, image pair the given image correspond to the given text or not. We train the discriminator on (correct text, correct image), (correct text, wrong image) and (wrong text, correct image) pairs so that it can discriminate between correct and in correct pairs[1] with more weight given to the error originated in training discriminator on (correct text, correct image) pair and less weight to the other two pair's error. The only difference with the training of a normal classification network is that the distribution of one of the classes is not static but changes over time (as the generator gets better at creating real-looking data).



```
GAN Training algorithm
Input: correct image i , correct text t , incorrect image $\hat{i}$ ,incorrect text $\hat{t}$
For i=1 to epochs do:
  For j=1 to num_batches do:
    $T = \varphi(t)$ {encoding correct text using rnn}
    $\hat{T} = \varphi(\hat{t})$ {encoding incorrect text}
    $I = \omega(i)$ {encoding correct image}
    $\hat{I} = \omega(\hat{i})$ {encoding incorrect image}
    $R_{loss} = max(0, \alpha - cos_{similarity}(I, T) + cos_{similarity}(I, \hat{T})) + max(0,$
                  $\alpha - cos_{similarity}(I, T) + cos_{similarity}(\hat{I}, T))$
    $F_G = G(Z, T)$ {G takes input noise(Z) and text(T) and gives image F}
    $F_{DW} = D(F_G, T)$ {D's input is Wrong(Generated) image,Correct text pair }
    $F_{DC} = D(I, T)$ { D's input is Correct image,Correct text pair }
    $F_{DF} = D(I, \hat{T})$ {D's input is Correct image,wrong text pair }
    $G_{loss} = \gamma(F_{DW}.logits)$ {calculating generator loss}
    $D_{loss} = \gamma(F_{DC}.logits) + ( \gamma(F_{DC}.logits) + \gamma(F_{DC}.logits) )/2$
    $ADAM(G, G_{loss})$ {updating G using the loss calculated}
   $ADAM(D, D_{loss})$ {updating D using the loss calculated}
   $ADAM(CNN\ RNN, R_{loss})$ {updating CNN,RNN using the loss calculated}
```

For the generator we back-propagate the gradients of the discriminator's output with respect to the generated image

i.e for (correct text, wrong image) pair. If we add these gradients to the generated image, it will make the image more realistic (from the discriminator's point of view). Instead, of adding these gradients to the image we will back-propagate these gradients further into all the weights that made up the generator so that the generator learns how to generate this new image.

GAN Training algorithm summarizes the training procedure in which we at first take correct, incorrect images and correct, incorrect texts and encode them using CNN and RNN respectively[4]. After encoding RNN loss is calculated based on how much similarity score is assigned to the correct text and image pair to correspond to the visual information conveyed by the text. After that generator generates an image that is fed to the discriminator, as this would correspond to the incorrect image and correct text pair. Then a pair of correct text, correct image and an incorrect text and correct image pair is fed to the discriminator and then the respective losses of the generator and discriminator are calculated and by using Adam optimizer gradient are used to update their weights as per the losses.

VIII. Experiments And Results

We performed our experiment on Oxford 102 flowers dataset. It consists of 7130 training images and our batch size was 64 so it did 115 iterations in each epoch. We trained our model for 800 iterations while only training RNN and CNN for the first 80 iterations. And after that training only the discriminator and the generator for remaining epochs.

We have used only 5 captions per image in order to constrain the amount of computation and training time required to train the model while still being able to generate image of acceptable quality, reducing the number of captions further reduces training time but the quality of images that were generated suffers significantly, so we decide 5 to be the limiting case where the quality of the images didn't suffer much and the time taken to train the model was also not much. But it can beincreasedupto10togetmuchbetterresults.

To further speed up the model's convergence we used glorot initializer to initialize weights[3] and have used learning rate decay in which we decay learning rate after a fixed number of iterations(here we have used 100) when we will be near the minima in loss landscape the model will oscillate in a much tighter region. The intuition behind it is that as the learning begins we can afford to take big steps but as the learning approaches convergence then having a smaller learning rate allows the model to take smaller steps towards minima.

In the dataset the dimensions of the images is 64*64*3. We have used beta1 to be 0.5 as the momentum for Adam [2] update. Dimension of noise is 1*512 and the noise vector is drawn from a normal distribution. We have used text feature dimension to be 128 and that to of word embedding to be 256andvocabsizetobe8000.

Images below represents the results obtained when we tested our model against text queries mentioned in the caption of the images.

Query 1: The flower shown has yellow anther blue pistil and yellow petals.

Query 2: This flower has petals that are white, and has dark lines

Query 3: The petals on this flower are pink with a dark center.

Query 4: This flower has a lot of small round blue petals.

Query 5: This flower is red color petals.

Query 6: The flower has dark black petals and the center of it is brown and has black pistil.

Query 7: The flower shown has green petals with dark center green anther green pistil.

Query 8: These white flowers have petals that start off white in color and end in a white towards the tips.

There is an absence of an proper standard evaluation metric for generative model. These models are judged by the quality of images generated by them which is done under human supervision. So in below image we show our results obtained on the above 8 queries.

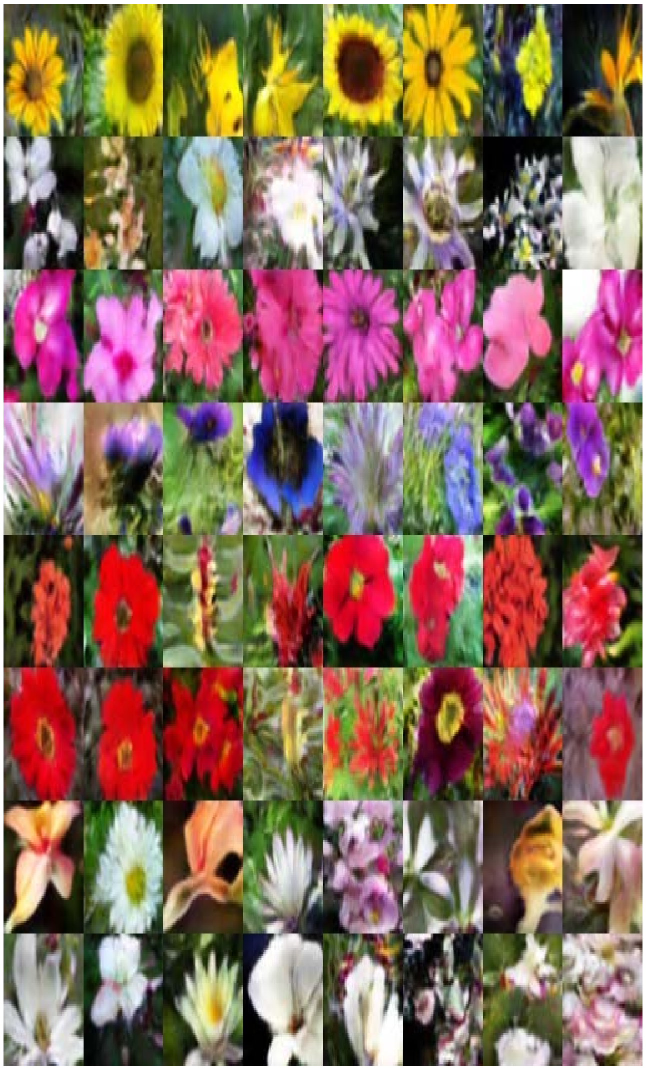


Fig 4. RGAN Results

IX. CONCLUSIONS

In this work we presented an text to image architecture that utilizes skip connections along with processes like learning rate decay to stabilize the training process and reach the convergence faster resulting in the generation of reasonable quality images conditioned on the visual information contained in a detailed text description. We showed that GAN conditioned on a text description can produces several images that fit the description correctly which is a natural result of multi-modality existing in GANs.

X. FUTURESCOPE

GANs produce quality outputs and provide a novel way to create generations similar to real-world scenarios. At their current stage, there is a lot that can be done in GANs in areas such as ensuring the variety of output and finding suitable ways to measure and rate the output, stabilize the learning process and new ways for the model to reach convergence faster. Research is needed to make GANs more controllable .As it can be seen in the results that the model is not perfect and it can be attributed to the fact that we have only taken five caption to restrict the compute and training time so in future work, we aim to further scale up the model to generate higher resolution images and add more types of text with attention also giving to the background part and also to the foreground so the overall quality of the generated image improves resulting in more sharp detailed realistic output images.

REFERENCES

[1] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran Bernt Schiele, Honglak Lee, "Generative Adversarial Text to Image Synthesis", University of Michigan, Ann Arbor, MI, USA ICML'16 Proceedings of the 33rd International Conference on International ConferenceonMachineLearning-Volume48Pages1060-1069

[2] Diederik P. Kingma. 2015. and Jimmy Lei Ba. Adam: a Method for Stochastic Optimization. International Conference on Learning Representations, pages1-13.

[3] "Understanding the difficulty of training deep feedforward neural networks",Xavier Glorot, Yoshua Bengio ; Proceedings of the Thir- teenthInternationalConferenceonArtificialIntelligenceandStatistics, PMLR 9:249-256,2010.

[4] Sergey Ioffe and Christian Szegedy "Batch Normalization: Accel- erating Deep Network Training",ICML'15 Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37 Pages448-456

[5] Radford, Alec, Luke Metz and Soumith Chintala. "Unsupervised Rep- resentation Learning with Deep Convolutional Generative Adversarial Networks."CoRRabs/1511.06434(2015).

[6] Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In NIPS,2014

[7] Ledig, Christian Theis, Lucas & Huszar, Ferenc & Caballero, Jose & Cunningham, Andrew & Acosta, Alejandro Aitken, Andrew & Tejani, Alykhan &Totz, Johannes &Wang, Zehan & Shi, Wen- zhe. (2017). Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. 105-114.10.1109/CVPR.2017.19.

[8] Reed, Scott & Akata, Zeynep & Lee, Honglak & Schiele, Bernt. (2016). Learning Deep Representations of Fine-Grained Visual De- scriptions. 49-58.10.1109/CVPR.2016.13.

[9] Zhang, Han Xu, Tao & Li, Hongsheng. (2017). StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks. 5908-5916.10.1109/ICCV.2017.629.

[10] Wang, Meng & Li, Huafeng & Li, Fang. (2017). Generative Adver- serial Network based on Resnetfor Conditional Image Restoration.

[11] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin

Riedmiller, Thomas Brox (2015) Discriminative Unsupervised Feature Learning with Exemplar Convolutional Neural Networks.

[12] Junhua Mao, Wei Xu & Yi Yang & Jiang Wang & Zhiheng Huang & Alan Yuille (2015). Deep Captioning With Multimodal Recurrent Neural Networks (M-Rnn), In: Proceedings of 3rd International Conference on Learning Representations, *ICLR 2015*, San Diego, CA, USA, May 7-9, 2015.

[13] Andrej Karpathy, Li Fei-Fei(2015) Deep Visual-Semantic Alignments for generating image descriptions. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3128-3137.