

Text-to-Image: ImageGAN

Soham Das (2019477), Jahnvi Kumari (2019469), Eeshaan Ravi Tiwari (2019465)

Indraprastha Institute of Information Technology, Delhi

{soham19477, jahnvi19469, eeshaan19465}@iiitd.ac.in

1 Introduction

1.1 Problem Statement

Generating realistic images that match the given textual descriptions is a challenging problem in computer vision. It has various applications including image inpainting (Fu et al., 2020), suspect's face generation (Jalan et al., 2020), medical imaging (Yi et al., 2019), etc. Recognizing that text is the most natural and convenient medium used for describing an image by human beings, we aim to develop a deep architecture and Generative Adversarial Networks (GAN) (Goodfellow et al., 2014) formulation to effectively bridge the advances in text and image modeling translating visual concepts from characters to pixels by converting text in the form of single-sentenced captions directly into one or more image that fits that description.

1.2 Motivation

There are many underlying challenges that lie in the face of solving the text to image problem such as the inability of resulting images to depict precisely what has been described by the textual information. This is because the space of possible images that match the given text descriptions can be enormously huge. Other issues include the inability of the model to generate images in desired sizes and the Cross-domain nature of the problem (Text to Image Generation), which requires careful treatment and expertise in both fields. However, significant improvements and new techniques' introduction in areas such as GANs boost the confidence of researchers to tackle some of the issues described above remarkably and generate realistic images with diverse conditions. We also aim to make significant contributions by picking up a few of these problems and trying to develop a solution for them from our side and make our work novel since the scope of modification in the GAN architecture is huge.

1.3 Project Pipeline Summary

Our proposed solution is mainly divided into 3 parts - text embeddings generation, Generator module, and Discriminator module. Starting from the text embedding generation task, firstly, we pass all the captions from our dataset to get their sentence-level embedding or representation using Google's Universal sentence encoder. Next is the Generator module where these generated sentence embeddings are passed as inputs. The generator module consists of series of convolution blocks and affine block which are described in the Methodology section. The generator uses these embedding along with the random noise to generate images. Lastly, we enter into the Discriminator module, where using the embeddings, the real image, and the generated image from the Generator, the discriminator decides whether the generated image is real or fake.

2 Related Work

Since a major part of our task is image generation, GANs have been an integral part of our study. Various modifications to GANs have been proposed and have helped make robust models for generating images from text input. In our literature survey, we have included details from six peer-reviewed publications. At the core of each such publication is a modified version of GAN that embeds information from text into the generated image. In (Reed et al., 2016b), the authors used a hybrid character-level convolutional RNN text encoder and a class-conditional GAN trained using a novel strategy for fine-grained image datasets like the Caltech-UCSD bird's dataset and the Oxford-102 Flowers dataset. They modified the discriminator (GAN-CLS) to become matching aware. Usually, at the beginning of training, the discriminator ignores the conditioning information and easily rejects samples from G because they do not look plausible. Once G has learned to generate plausible images, it must

also learn to align them with the conditioning information, and likewise, D must learn to evaluate whether samples from G meet this conditioning constraint. They simplified this learning process by supplying an additional input of real images with mismatched text which the discriminator must learn to score as fake. Hence, they optimize the image-text mapping in addition to image realism from the beginning itself. They also make GAN-INT wherein they introduce a generator objective to interpolate between text embeddings. These additional text embeddings help discriminators learn to predict whether image and text pairs match or not. The text embedding mainly covers content information and typically nothing about style, therefore, to generate realistic images, GAN must learn to use noise sample z to account for style variations. A style encoder has been used to disentangle style (background and pose) for this purpose. StackGAN (Zhang et al., 2017) generates photo-realistic images conditioned on text descriptions. The Stage-I GAN sketches the primitive shape and basic 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 photorealistic details. The Stage-II GAN is able to rectify defects and add compelling details with the refinement process. Samples generated by StackGAN are more plausible than those generated by existing approaches. Discontinuity in the data manifold is caused by transforming the text embedding non-linearly (as in the above paper where text encoding was passed through the MLP layer to convert it into 128 dimensions) into a high dimension latent variable space for the generator’s learning. To mitigate this problem, they introduce a conditioning augmentation technique to produce more conditioning variables for the generator, which yields more training pairs given a small number of image-text pairs (Reed et al., 2016b) used interpolation for the same cause). This paper uses (Reed et al., 2016b)’s matching aware discriminator (GAN-CLS) for both stages. The architectures of both the stages are the same with the exception that the second stage samples from the low-resolution image output of the stage 1 GAN whereas stage 1 GAN samples from the normal distribution. AttnGAN (Xu et al., 2018) deals with the issue of conditioning GAN only on the global sentence vector. It claims that this global vector lacks important

fine-grained information at the word level, and prevents the generation of high-quality images. Therefore, they propose a novel attention model that enables the generative network to draw different subregions of the image conditioned on words that are most relevant to those sub-regions. They also introduce a Deep Attentional Multimodal Similarity Model (DAMSM) to learn two neural networks that map subregions of the image and words of the sentence to a common semantic space, thus measuring the image-text similarity at the word level to compute a fine-grained loss for image generation. To overcome the problem of high computation costs due to attention and stacking GANs, (Hu et al., 2021) proposed a novel one-stage framework called SSA-GAN. It introduces a Spatial Semantic Aware (SSA) block to fuse the text and image features effectively and deeply by predicting semantic masks to guide the learned text-adaptive affine transformation at the pixel level. The semantic mask predictor is trained in a weakly supervised way, such that no additional annotation is required and this block is the potential to be applied to other T2I datasets. Uses bidirectional LSTM as text encoder, pre-trained on Deep Attentional Multimodal Similarity Model (DAMSM) loss. To address the issues of computational cost due to stacking, attention, etc., (Tao et al., 2020) proposed a Deep Fusion Generative Adversarial Network (DF-GAN) wherein they replaced the stacked backbone with a one-stage backbone composed of hinge loss and residual networks. To ensure text-image semantic consistency, they proposed a novel target aware discriminator composed of Matching-Aware Gradient Penalty (MA-GP) and a Deep text-image Fusion Block (DFBlock) to effectively fuse the text information into image features. (Zhu et al., 2019) proposes DM-GAN (Dynamic Memory Generative Adversarial Network) address the two issues. One is the dependence of the quality of the generated image on that of the initial image, and the other is that in the previously existing methods, the importance of a word in a sentence was not determined using the context of the image content. To fix the first problem, they propose to add a memory mechanism that can generate high-quality images even with unfavorably generated initial images. For the second issue, they introduce a memory writing gate that can pick up relevant words from the caption in the context of the initial image. They also propose a response gate to adaptively fuse image and memory

information, instead of direct concatenation.

3 Methodology

3.1 Dataset Details

We have used the flower-102 category dataset, consisting of 102 flower categories. Each class consists of between 40 and 258 images. Since our task required the textual description of the images and the original version of the dataset didn't have the captions associated with the images describing the characteristics of the flowers present in the image, we have used another version of this dataset from (Reed et al., 2016a). We have used the images and the captions from this data to make our embedding using the pretrained universal sentence encoder from Google.

3.2 Sentence-level Embedding Generation

To get the sentence-level embeddings of flower's images we have used the pretrained Universal Sentence Encoder model from Google which takes the image captions as input and gives the sentence level embedding of length 512 as output. The universal sentence encoder has 2 underlying encoding versions- the transformer version and the Deep Averaging Network(DAN) version. Since the transformer version is designed for better performance, we have used the transformer version for our purpose.

3.3 Affine Block

To improve the quality of the generated image, we further fuse textual information into the image. This is done using Affine Transformation (Tao

et al., 2020) (Figure 1). Given an input feature map X (belongs to) $R^{B \times C \times H \times W}$ and an input sentence vector e , two parameters $\gamma \in R^{B \times C}$ and $\theta \in R^{B \times C}$ are estimated using MLPs. The γ and β parameters are used to scale and shift each channel of the input feature map respectively.

$$AFF(x_{ij}|e) = \gamma_{ij} \cdot x_{ij} + \beta_{ij}$$

, where x_{ij} is the i th channel of the feature map of the j th sample in the batch. γ_{ij} and β_{ij} are the parameters corresponding to x_{ij} . This transformation does not change the dimensions of the input feature map. (Tao et al., 2020) applies ReLU activation between two affine layers to introduce non linearity in the fusion of textual information, and use only interpolation for increasing the H and W dimensions. We omit this in our implementation as we use transposed convolution and non linear activations for our upsampling.

3.4 Model Architecture

In generator network, we first sample a noise vector z (belongs to) $R^{100} \sim N(0,1)$, and create sentence embedding of the caption using our pretrained encoder. The dimension of the embedding is reduced to 128 using a fully connected layer followed by batch normalization and LeakyReLU activation, and this vector is concatenated to the noise vector. This results in a 228 length vector, which can be treated as a feature map X (belongs to) $R^{B \times 228 \times 1 \times 1}$. To implement the upscaling of the last two dimensions, we use upsampling blocks (UpBlockStart, UpBlock, UpBlock_Tanh) and interpolation, as shown in Figure 2. The dotted connections from the embedding projection to the upblocks represent the optional affine transformations after upsampling. In discriminator network, we get an image and a

Figure 1: Illustration of affine transformation (Tao et al., 2020)

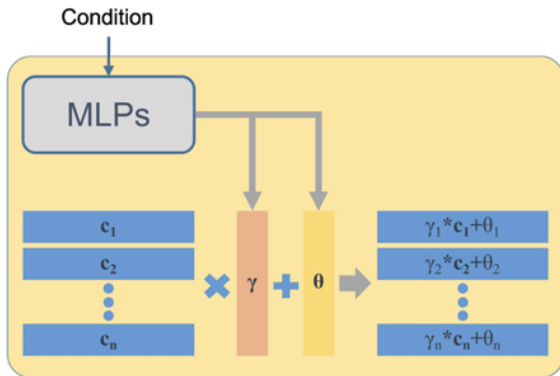
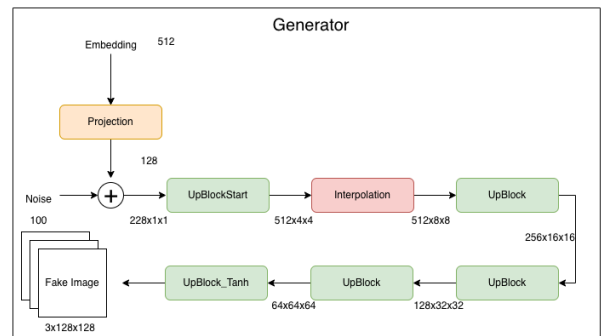


Figure 2: Our generator architecture



caption as input. We reduce the dimension of the text embedding to 128 using a fully connected layer followed by batch normalization and LeakyReLU activation. We reduce the image spatial dimension to 4x4 using downsampling blocks (DownBlockStart and DownBlock) and interpolation. Now, we replicate the text embedding projection 16 times, arrange it into a 4x4 spatial dimension, and concatenate it into the feature map. Finally, a stride 1 convolution (DownBlockSig) is used to get a sigmoid output of 1x1 spatial dimension and 1 channel. The steps are shown in Figure 3.

3.5 Loss Functions

We derive our loss functions from (Reed et al., 2016b). The objective of the generator is to create an image that the discriminator cannot tell apart from a real image corresponding to the given caption. The discriminator must be able to tell between an unrealistic image, a realistic image that does not correspond to the given text, and a realistic image that does. To separate these, the discriminator loss comprises of three terms as shown in Figure 4. s_r , s_w , and s_f refer to the score calculated by discriminator when associating a fake image with the given caption, a real image with a wrong caption, and a real image with its corresponding caption respectively. The discriminator calculates the probability of the image being fake. Therefore, the discriminator will try to minimize s_r , and maximize s_w and s_f . On the other hand, the generator will try to minimize s_f . One sided label smoothing was added in the discriminator loss, and feature matching in the generator loss to mitigate overconfidence of discriminator and improve stability of the model (Salimans et al., 2016). Apart from that, we also

add the pixel-wise L1 difference between the real and fake images in the generator loss, so that certain pixels of the generated image also learn textual features.

4 Experiments

In the first experiment, we did not add any affine layer in all the upsampling blocks. Thus, after concatenating the noise vector with the text embedding from the universal sentence encoder, the data is fed into 4 such upsampling blocks in the generator. Next, we hypothesized that adding affine transformations at different stages of the generator can lead to different results. To prove this and observe the nature of difference, we added an affine layer in only the first two upsampling blocks in the second experiment, and only the last two in the third experiment. Lastly, we added an affine block in all the upsampling blocks which gave the clearest picture and a 3 times better inception score than a non-affine generator. We ran all the experiments for 50 epochs each and obtained the best performing model by running the fourth experiment for 101 epochs in Figure 6. Other than that, we tried two different learning rates of 0.002 and 0.02 and ran the model for 30 epochs on the 0.02 value. The results are listed in Table 1.

5 Results and Analysis

We rank the models based on their Inception Scores on the test set. Inception Score or IS is an objective metric for evaluating the quality of generated images, specifically synthetic images output by generative adversarial network models. IS is the exponential of the KL divergence which measures the

Figure 3: Our discriminator architecture

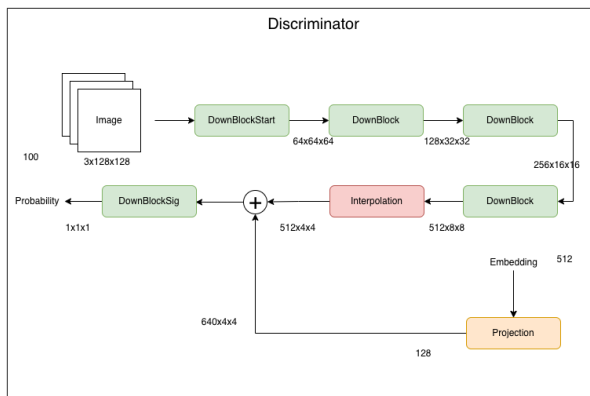
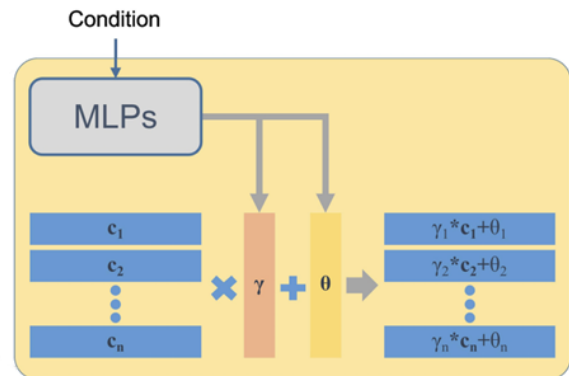


Figure 4: Training algorithm in (Reed et al., 2016b)



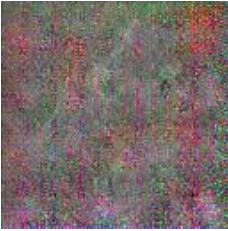



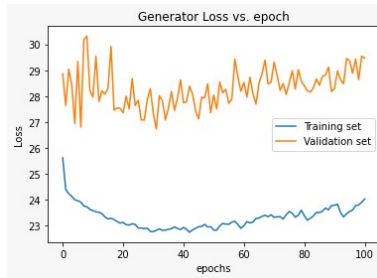
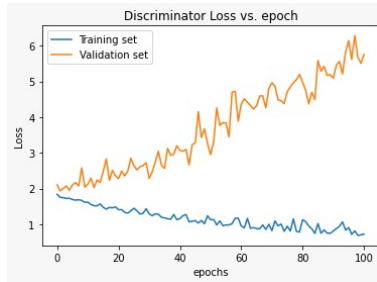
| Upsampling block | Inception Score | Generated Image |
|----------------------------|-----------------|--|
| No affine | 1.001 |  |
| Affine in last two blocks | 2.876 |  |
| Affine in first two blocks | 2.773 |  |
| All affine | 2.975 |  |

Table 1: Results on our experiments. The images are generated on the caption "Red flower with Yellow Pistils".



(a) Generator loss



(b) Discriminator loss nodes

Figure 5: The loss curves for all affine upsampling blocks for 100 epochs.

similarity between two probability distributions (in our case, the "distinctness" the images). Higher IS denotes better quality. Unlike the images generated by GANs which have an interpolation layer after the generator, our images with the increased size are not blurred. This means we have improved interpolation by adding it to an intermediate generator step. The loss plots starts stagnating after 20 epochs Figure 5, hence we conclude that the model overfits. However, it's performance still improves as it was able to learn useful features after 51 epochs too.

6 Individual Contribution

We have mutually collaborated and worked equally on code. The individual tasks that were divided to take charge are as follows -

Jahnvi Kumari: Interpolations, that is Increasing output's image dimensions, code for inference time, experimentation.

Soham Das: Fusing of image embedding into Generator's convolution blocks, experimentation

Eeshaan Ravi Tivari: Creation of image embeddings using Google's universal sentence encoder, and hdf5 file generation for training the model.

Figure 6: Output image for caption "Red flower with yellow pistils" for all-affine model trained on 100 epochs.



References

- Guokai Fu, Chen Diao, Wanli Xue, and Shengyong Chen. 2020. Noise-regression gan for image inpainting and multiple generation. In *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, pages 383–387. IEEE.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.
- Kai Hu, Wentong Liao, Michael Ying Yang, and Bodo Rosenhahn. 2021. Text to image generation with semantic-spatial aware gan. *arXiv preprint arXiv:2104.00567*.
- Harsh Jaykumar Jalan, Gautam Maurya, Canute Corda, Sunny Dsouza, and Dakshata Panchal. 2020. Suspect face generation. In *2020 3rd International Conference on Communication System, Computing and IT Applications (CSCITA)*, pages 73–78. IEEE.
- Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. 2016a. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 49–58.
- Scott Reed, Zeynep Akata, Xinchun Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. 2016b. Generative adversarial text to image synthesis. In *International conference on machine learning*, pages 1060–1069. PMLR.
- Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016.

Improved techniques for training gans. *Advances in neural information processing systems*, 29.

Ming Tao, Hao Tang, Songsong Wu, Nicu Sebe, Xiao-Yuan Jing, Fei Wu, and Bingkun Bao. 2020. Df-gan: Deep fusion generative adversarial networks for text-to-image synthesis. *arXiv preprint arXiv:2008.05865*.

Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. 2018. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324.

Xin Yi, Ekta Walia, and Paul Babyn. 2019. Generative adversarial network in medical imaging: A review. *Medical image analysis*, 58:101552.

Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. 2017. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915.

Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. 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*, pages 5802–5810.