Facial Image Generation From Textual Descriptions

Vidit Jain 2017370
Arjun Garg 2017138
Neha Bhairavi Prakriya 2017168

1. Abstract

Text to vision correlation has been a growing source of interest for NLP researchers around the world. Particular efforts in the area include scene descriptors for environmental surroundings which have shown promising results. Face descriptors however, remain a complicated yet interesting task with applications in fields ranging from generating videos from textual descriptions, sketching criminals based on textual description of their face, as well as in aiding the understanding of complicated textbooks for students with learning difficulties.
In this paper we present three models namely CVAE, DCGAN and CVAE-GAN implemented and tuned for text-to-face generation.

2. Introduction

NLP researchers around the world have thrust their efforts into the text-to-image applications for the exciting opportunities they offer. This task, if perfected can be used for anything from being able to generate entire cricket matches based on the commentary to facilitating learning for children. However, scene descriptions are difficult to achieve due to the complexities involved in their design and implementation.
Most existing scene descriptions have been limited to generating simple flowers from text. This does not explore the complexities involved in generating faces with variable aspects like skin colour, hair texture, eye colour etc.

2.1. Dataset exploration

The dataset we are using is Face2Text. This dataset consists of 50000+ labeled samples. The dataset contains images from CelebA and its textual descriptions. The images cover a large range of pose variations, background clutter, diverse people and provides a large quantity of images and rich annotations. The descriptions mostly consist of visual content,however, many descriptions are based on emotional and inferential content. Each image in the dataset has a size of 218x178. Some samples from the dataset can be seen in Figure 1.

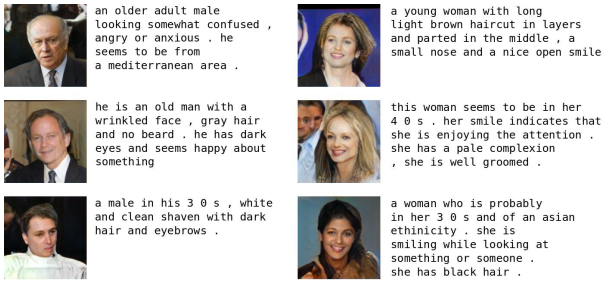

Figure 1. Samples from the Dataset

3. Related Work

The use of facial images has been limited to image recognition tasks and very little exploration has gone into text-to-face generation. Efforts in text-to-image generation have been limited by the lack of well attributed data. Datasets like MS COCO, Flickr do not contain image captions at a fine granularity as needed by many performance critical image generation applications. Also, text and images being in different modalities contributes to the complexity of designing robust models.
Despite these limitations, the recent improvements in generative learning have enabled works like [1] which describe the use of a GAN for generation of flower images using the Oxford-102 flowers dataset [5], bird images using the Caltech ucsd birds 200 [3], and general objects using the MS COCO dataset [6].

4. Preprocessing

We randomly sampled 10,000 images from the dataset. Due to crunch of resources, the images were scaled down from 218x178 to 64x64. Out of the total, 7500 images were used for training the models and the rest 2500 were used for testing purposes. Corresponding captions for each image were also loaded. We used **Skip-Thought Vectors** [2] to encode the input captions to a 4800 dimension vector. We use the pre-trained model provided by the authors. The authors use an encoder-decoder model to generate skip-

thought vectors. The encoder maps the sentence to a vector and the decoder generates surrounding sentences from the vector. These vectors have obtained quite good results for image retrieval tasks on the MS COCO dataset [6].

5. Methodolgy

We have used three models for text-to-image generation namely Conditional Variational Autoencoders (CVAE), Deep Convolutional Generative Adversarial Networks (DCGAN) and CVAE-GAN, a combination of the CVAE and GAN which has gained prominance in recent works. In the following sections, we describe each of the three models in depth.

5.1. GAN-CLS Algorithm

The traditional way to train a conditional GAN is to view text-image pairs as joint observations and train the discriminator to judge the pairs as real or fake. However, such type of conditioning is credulous as the discriminator has no way to know whether the training images match the embedded caption. Thus training a matching aware discriminator sees three kinds of inputs: real image with matching text, generated image with matching text and real image with wrong text. Thus it learns to identify a mismatch between images and text. By learning such a mapping, the discriminator can provide better training of the discriminator. Algorithm 1 summarises the training procedure [1].

Algorithm 1 GAN-CLS training algorithm with step size α, using minibatch SGD for simplicity.

Input: minibatch images x, matching text t, mismatching $\hat{t}$, number of training batch steps S
for $n = 1$ **to** S **do**
  $h \leftarrow \varphi(t)$ {Encode matching text description}
  $\hat{h} \leftarrow \varphi(\hat{t})$ {Encode mis-matching text description}
  $z \sim \mathcal{N}(0,1)^Z$ {Draw sample of random noise}
  $\hat{x} \leftarrow G(z,h)$ {Forward through generator}
  $s_r \leftarrow D(x,h)$ {real image, right text}
  $s_w \leftarrow D(x,\hat{h})$ {real image, wrong text}
  $s_f \leftarrow D(\hat{x},h)$ {fake image, right text}
  $\mathcal{L}_D \leftarrow \log(s_r) + (\log(1-s_w) + \log(1-s_f))/2$
  $D \leftarrow D - \alpha\partial\mathcal{L}_D/\partial D$ {Update discriminator}
  $\mathcal{L}_G \leftarrow \log(s_f)$
  $G \leftarrow G - \alpha\partial\mathcal{L}_G/\partial G$ {Update generator}
end for

5.2. CVAE

Conditional Variational Autoencoder [7] introduced in 2015 has obtained excellent results and is among the state of the art approaches to generative modeling thus forming a good baseline model for the proof-of-concept for text-to-face generation. The CVAE evolved from the classic autoencoder designs which involved a two-step process of encoding input data into the latent space and regenerating the input data from these feature representations. While the Variational Autoencoder (VAE) introduced the capability to generate synthetic data using the probability distribution of the data. CVAEs bring the ability to generate synthetic data for a specific label. As can be seen in the figure below, in case of a CVAE, the prior distribution and the output distribution are different from that used in case of VAEs.

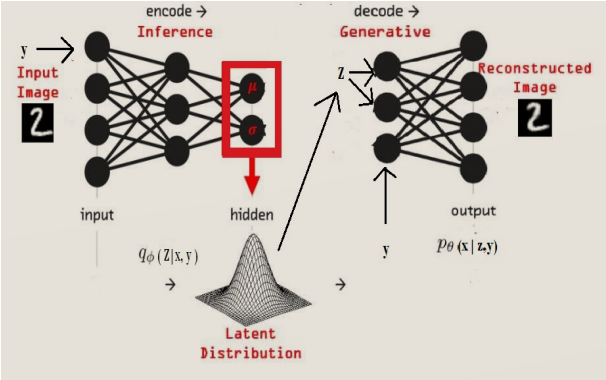

Figure 2. CVAE

5.3. DCGAN

Deep Convolutional GAN is one of the most popular and successful network design for GANs. It comprises of convolution layers and fully connected layers (only at the extremes). It uses strided and transposed convolutions for downsampling and upsampling. The generator takes noise drawn from a random distribution and through a series of transposed convolution layers generates an image. The discriminator, on the other hand, uses strided convolutions to map the input image to a 1-dimensional output and predict whether the input image is real or fake.

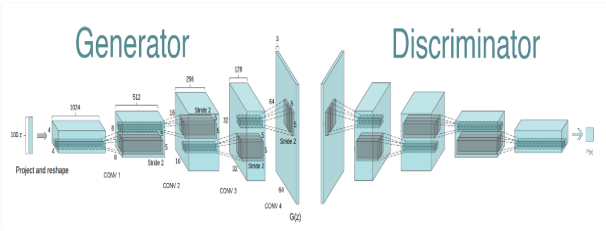

Figure 3. DCGAN

5.4. CVAE-GAN

This model [9], introduced in 2017, draws from the successful performance of GANs and VAEs with a modified objective function for the generator. Since images generated from the VAE are often blurry, the discriminator can classify them as fake. Therefore, the objective function of the generator is modified to incorporate minimising of the *l2* distance between the real image and the generated image. As in figure 4 the architecture of the CVAE-GAN consists of the following:

- Encoder which maps input x to a latent variable z
- A generator G which generates image x' from z
- A discriminator D which distinguishes the images generated as real/fake and,
- A classifier network C to measure the class probability of the data.

6. Design Choices

6.1. CVAE

Architecture. The encoder consists of 4 convolution layers with kernel sizes of (5*5) and number of filters going from 64, 128, 256 to 256. The stride of the filters is kept at 2 to reduce the input size by half. This is followed by a FC layer. The dimension of the latent vector is 128. This is followed by the decoder which consists of (5*5) size filters with each layer consisting of 256, 128, 64 and 3 filters.
Hyperparameters. The model has been trained for a total of 150 epochs with 7500 images. The optimiser used is Adam and the learning rate has been set to 0.0002. The momentum parameter βhas been taken as 0.5.

6.2. DCGAN

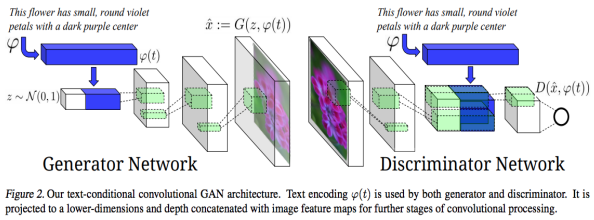

Figure 2. Our text-conditional convolutional GAN architecture. Text encoding $\varphi(t)$ is used by both generator and discriminator. It is projected to a lower-dimensions and depth concatenated with image feature maps for further stages of convolutional processing.

Figure 4. DCGAN architecture

Architecture. We sample the input noise *Z=U(-1,1)* of dimension 100 and encode the input caption to a 4800 dimension vector using the skip-thought encoder. The encoded caption is reduced to a 256 dimension vector using a fully connected layer followed by a leaky-ReLU activation. The noise and text-encoding are concatenated to form a 356 dimensional input to the generator.
The generator is a deconvolution network with four transposed convolution layers with kernel size=5, same padding and stride=2 followed by batch normalisation and leaky ReLU activation except the last layer which has tanh activation.
The discriminator is a convolution network with five strided convolution layers with kernel size=5, same padding and stride=2 except for the last layer with stride=1 followed by batch normalisation and leaky ReLU activation. The encoded caption and a 64x64x3 image serve as input to the discriminator. The 4800 dimensional encoded caption vector is reduced to a 256 dimensional vector. The first four convolution layers use a stride of 2 to reduce the input size by half and double the number of input channels. After the 4th convolution layer, the 256 dimensional text encoding vector and the convolved image are depth-wise concatenated and passed to the 5th convolution layer. The output of the final convolution layer is passed through a sigmoid activation to generate a confidence score between (0,1).
Hyperparameters. The model was trained for 200 epochs with 7500 images. Learning rate of the discriminator was set to 0.0001 and for the generator was 0.0002. We used the Adam optimiser for both the generator and discriminator with β_1=0.5.

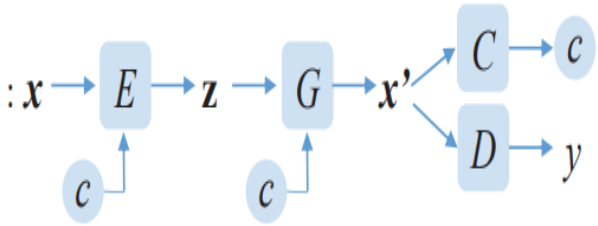

Figure 5. CVAE-GAN

6.3. CVAE-GAN

Architecture. The CVAE-GAN follows the architecture shown in Figure 4. The encoder has four convolutional layers with the number of filters ranging from 128, 256, 256, and 512. The kernel size is (5*5) and the stride is 2. This is followed by a dense layer. The decoder consists of four convolutional layers with filters ranging from 128, 256, 256, and 512. The discriminator has the same structure as the encoder varying only in the activation function used. In case of the encoder, ReLu activation followed by linear activation is used. In contrast, the discriminator uses ReLu followed by Sigmoid activation. The classifier too only differs in the activation functions used namely ReLu and

Softmax.

Hyperparameters. The model has been trained for a total of 70 epochs with 7500 images. The optimiser used for the generator, discriminator, encoder and classifier is Adam and the learning rate has been set to 0.0002. The momentum parameter βhas been taken as 0.5. The number of parameters in each model are shown in Table 1.

Model	CVAE	DCGAN	CVAE-GAN
Parameters	10.7M	23.8M	56.6M

Table 1. Number of parameters in each model.

7. Evaluation Metrics

Text-to-image synthesis utilize the Inception score as the criterion. To evaluate the networks, a pre-trained Inception v3 model is used to calculate the scores. A pre-trained Inception v3 model is available for the Caltech-UCSD Birds 200 (CUB-200) dataset [3]. However, we were not able to find any pre-trained Inception model for the CelebA dataset. Hence we turned towards other similarity metrics like the face semantic distance (FSD) and the Face Semantic Similarity (FSS) scores. We utilised the FaceNet model [4] to extract features from both the ground-truth and the corresponding generated faces to calculate these metrics.

$$FSS = \frac{1}{N}\sum_{i=0}^{N} cos(FaceNet(G_i) - FaceNet(T_i))$$

$$FSD = \frac{1}{N}\sum_{i=0}^{N} |FaceNet(G_i) - FaceNet(T_i)|$$

where *FaceNet()* means using the pre-trained FaceNet model to extract features from an input image, G_i means one of the predicted faces and T_i means the corresponding ground-truth face. A lower FSD and higher FSS score mean the generated face images are more similar to the ground truth. The final results are given in Table 2.
We also visually inspected the images produced in the same epochs across the models to determine the performance of the models as well.

8. Results: Loss Plots, Visualisations, FSS & FSD scores

8.1. CVAE

Loss plot for the CVAE is given in the figure below:
The reconstructions by the model before training, after 10 epochs and after 50 epochs is given in the figures below.

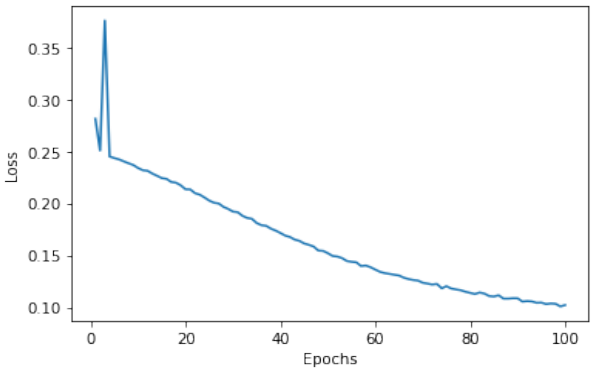

Figure 6. CVAE loss plot

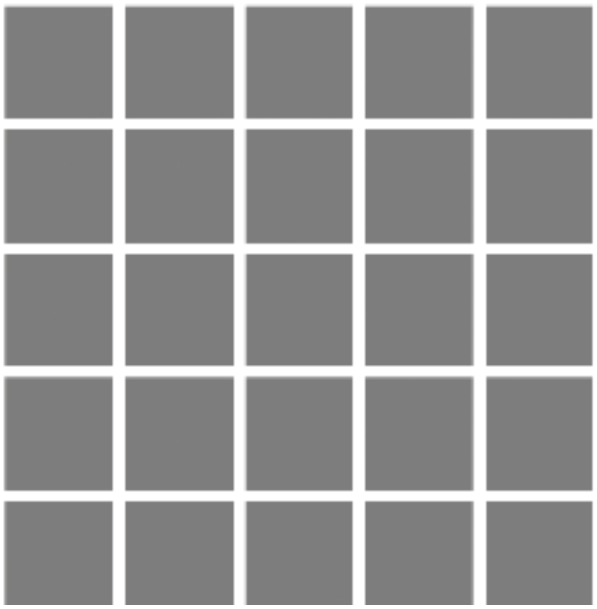

Figure 7. CVAE reconstruction before training

8.2. DCGAN

The generator and discriminator loss plots for DCGAN, generated image samples at the 0th, 50th and 100th epoch are given below and images generated by test captions are given below:

8.3. CVAE-GAN

The generator and discriminator plots for the CVAE-GAN can be found in figure 15. Further, reconstructions for 0,35 and 70th epoch can be found.

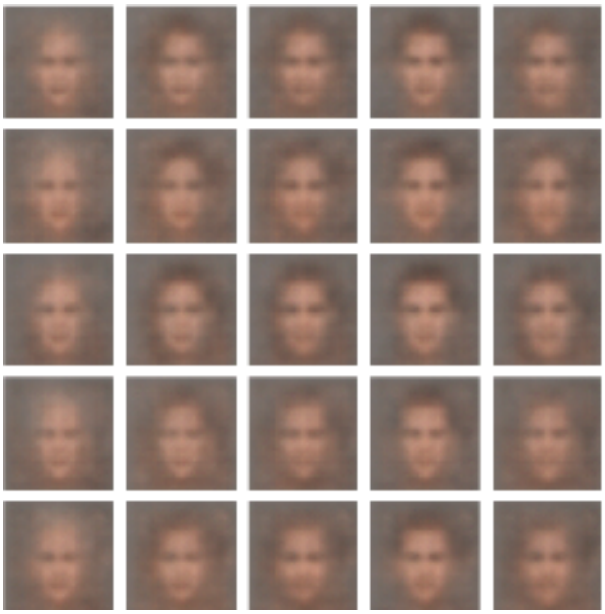

Figure 8. CVAE reconstruction after 10 epochs

Figure 9. CVAE reconstruction after 50 epochs

8.4. Scores

Model	CVAE	DCGAN	CVAE-GAN
FSD	81.22	71.91	**67.61**
FSS	0.218	0.258	**0.324**

Table 2. FSD and FSS scores for all models

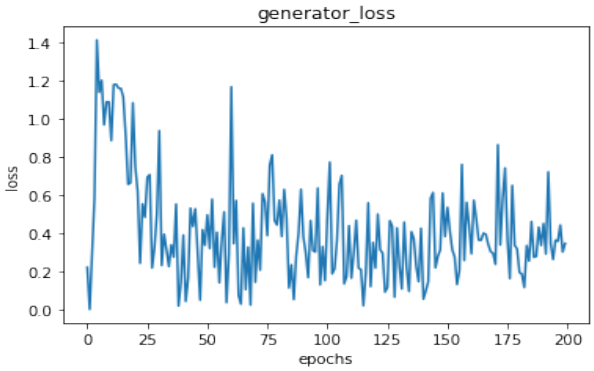

Figure 10. DCGAN Generator loss plot

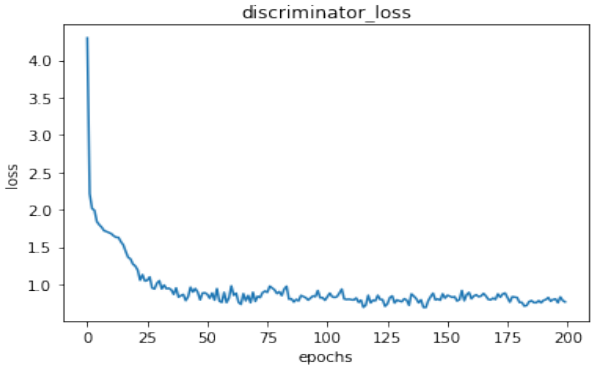

Figure 11. DCGAN Discriminator loss plot

Figure 12. DCGAN reconstructions at 0th epoch

Figure 13. DCGAN reconstructions at 50th epoch

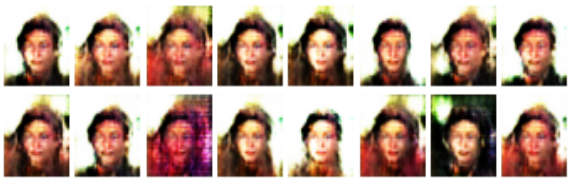

Figure 14. DCGAN reconstructions at 100th epoch

Figure 15. DCGAN generated image

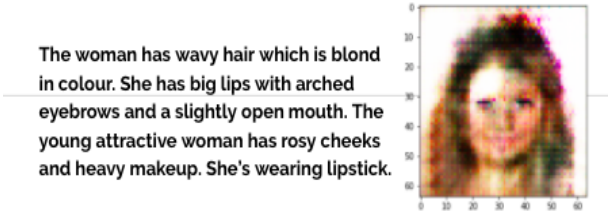

Figure 16. DCGAN generated image

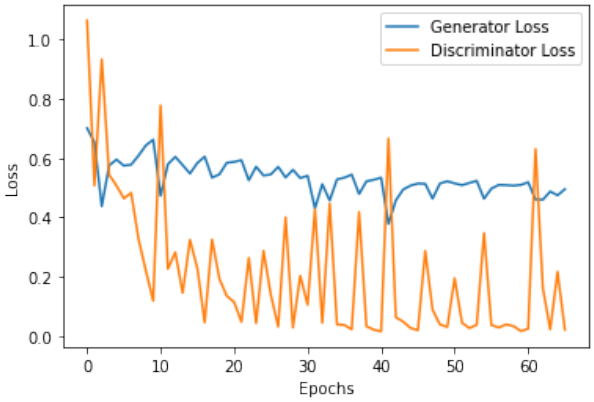

Figure 17. Loss Plots for CVAE-GAN

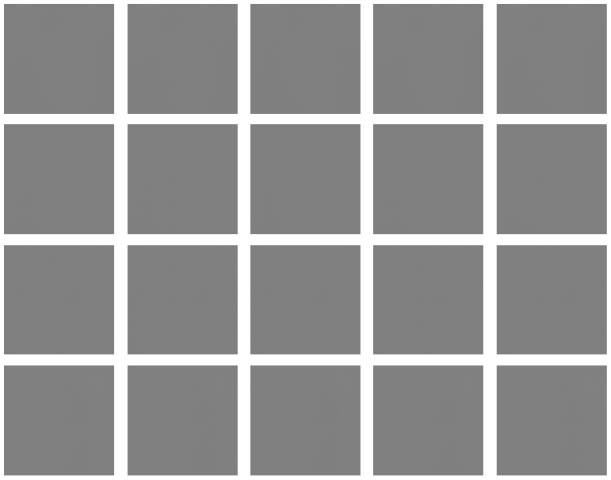

Figure 18. Reconstructions at 0th epoch for CVAE-GAN

9. Analysis

We trained a CVAE model for 150 epochs and from the loss plot we can observe that the conditional variational autoencoder is trained correctly. However, upon looking at the reconstructions we observe that the generated images are blurry and lack detail. Moving on, we explored GANs for the task of text to face synthesis. We trained a DCGAN model using the GAN-CLS algorithm mentioned above. From the loss plots we can see that the model has been trained fairly well. The image reconstructions were better than the previous model as it included good detail. We implement another model, CVAE-GAN that uses the capabilities of both GANs and VAEs. The model was pretty heavy so we could train it only for 70 epochs. As we can observe through the results in Table 2 in the previous section, CVAE-GAN performs the best in terms of the FSS and FSD scores. This is expected as the CVAE-GAN uses the best components of the the VAEs and the GANs. It alleviates the problem of vanishing gradients and also avoid mode collapse. Also, after looking at the loss plots and the image reconstructions, the results look promising.

10. Challenges Faced

- Generative models we used were pretty heavy so were difficult to train on Google Colab.
- Stabilising the training for GANs.

11. Deliverables

Initially, we had proposed on implementation of three models- DCGAN, ProGAN and StackGAN. However, due

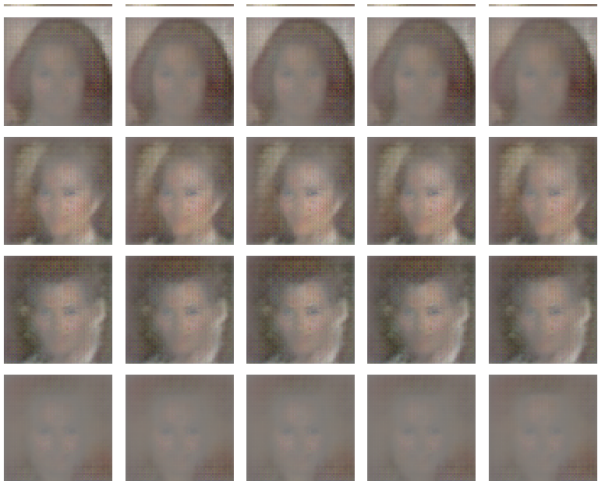

Figure 19. Reconstructions at 35th epoch for CVAE-GAN

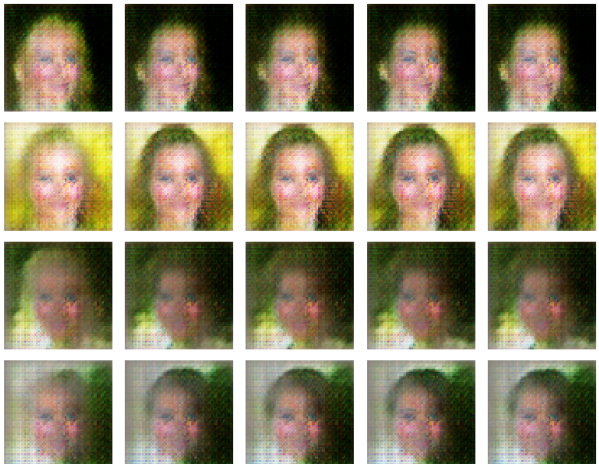

Figure 20. Reconstructions at 70th epoch for CVAE-GAN

Figure 21. Reconstructions at the 50th epoch

to the pandemic and unavailability of resources, our deliverables were readjusted to providing a proof of concept for the problem by implementing a simple CVAE. Our accomplishment of the project has not only been limited to the readjusted deliverables. We ended up implementing three models i.e. CVAE for proof of concept, DCGAN as our baseline model, and CVAE-GAN as our advanced model. These accomplishments are comparable to our initially proposed objectives.

12. Conclusions

In this paper, we propose and analyse the use of three generative models for text to image generation namely CVAE, DCGAN and CVAE-GAN. Through the course of this study, we compared the visual representations obtained from these models as well as the FSD and FSS scores obtained. We found that CVAE-GAN performed the best after which was DCGAN and then CVAE.

13. References

[1] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, Honglak Lee. Generative Adversarial Text to Image Synthesis. 2016. [2] Kiros, Ryan, et al. "Skip-thought vectors." Advances in neural information processing systems. 2015.
[3] Welinder P., Branson S., Mita T., Wah C., Schroff F., Belongie S., Perona, P. "Caltech-UCSD Birds 200". California Institute of Technology. CNS-TR-2010-001. 2010.
[4] Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
[5] M. Nilsback and A. Zisserman, "Automated Flower Classification over a Large Number of Classes," 2008 Sixth Indian Conference on Computer Vision, Graphics Image Processing, Bhubaneswar, 2008, pp. 722-729, doi: 10.1109/ICVGIP.2008.47.
[6] Lin, Tsung-Yi, et al. "Microsoft coco: Common objects in context." European conference on computer vision. Springer, Cham, 2014.
[7]Sohn, Kihyuk, Honglak Lee, and Xinchen Yan. "Learning structured output representation using deep conditional generative models." Advances in neural information processing systems. 2015.
[8]Bao, Jianmin, et al. "CVAE-GAN: fine-grained image generation through asymmetric training." Proceedings of the IEEE International Conference on Computer Vision. 2017.

14. Individual Contributions

Arjun Garg: CVAE-GAN
Neha Prakriya: CVAE
Vidit Jain: DCGAN