

# Using Generative Adversarial Networks for Textile Design Generation

Shams Arfeen

Department of Computer Science  
Habib Univeristy - Karachi, Pakistan  
sa05169@st.habib.edu.pk

Omema Ahmed

Department of Computer Science  
Habib Univeristy - Karachi, Pakistan  
oa04320@st.habib.edu.pk

Muhammad Munawwar Anwar

Department of Computer Science  
Habib Univeristy - Karachi, Pakistan  
ma04289@st.habib.edu.pk

**Abstract**—Deep generative models have recently played an important role in the growth of artificial intelligence in a variety of fields that need creativity. Despite its success, the generation of textile patterns is still largely unexplored. One possible explanation is that there is no publicly available data set large enough to train models on. However, research in this area can be useful because fashion designers can take up to several weeks to form a new design as they need inspirations and rely on a human’s creativity, but a trained GAN model can generate thousands of designs per day without the need of any human intervention. Furthermore, the use of GANs to generate fabric patterns from traditional Pakistani culture is an untapped field of research. In this research, we will be using DCGANs to generate fabric designs that are representative of Pakistani culture. In addition, we will compare the performance of DCGAN’s with StyleGAN and Variational Autoencoders. (VAE)

**Index Terms**—Generative Adversarial Network, Textile Patterns, Style Generative Adversarial Network, Pakistani Culture

## I. INTRODUCTION

Deep generative models have been crucial in the growth of artificial intelligence in a variety of disciplines that need creativity. One of the most extensively used generative models is generative adversarial networks (GAN) [5]. Generative adversarial networks are generative models that consists of one or more neural networks. The generator and the discriminator are what make up a of a neural network, where the former generators samples, starting from a random noise and the latter checks and classifies that image as either fake or real. The role of a generator is to minimize the probability of the discriminator to classify an image as fake. At the end of this process, the generator should be capable enough to produce images that are not distinguishable from the original dataset. Many models have been proposed to stabilize the training and improve the image quality.

Many disciplines are currently adopting GANs to provide high-quality outcomes for a range of tasks. GANs have also been widely used in computer vision for problems such as image-to-image translation, text-to-image translation, and super-resolution [15], [17], [12], [10]. GANs have also been utilised for image retrieval, the generation of synthetic data for deep learning tasks, image restoration, and surveillance.

Fashion is the most recent sector to be transformed by computer vision. The industry is now benefiting from break-

throughs such as neural style transfer [3], which utilises CNN to create synthetic customized clothes, and virtual try-on systems, which can generate a realistic image of the person with the desired clothes and posture. In the fashion business, generative models, namely GANs, are widely used. There are other applications such as producing model photos with customized clothing, SwapNet is an algorithm that transfers clothes from one person’s image to another irrespective of body location and shape while using GANs [14].

Textile industry has been the rampart of Pakistan’s economy. It contributes more than 60% to the total export earnings of the country, accounts for 46% of the total manufacturing and provides employment to 38% of the manufacturing labor force and 9% of GDP [1]. However, fashion design in Pakistan is more or less an entirely manual process, with little to no automation involved. It requires hours of manual labor to design a prototype pattern, and it goes through multiple iterations and feedback loops to get it finalized. The pattern itself also has to be designed from scratch, and its quality is entirely contingent upon the human designer.

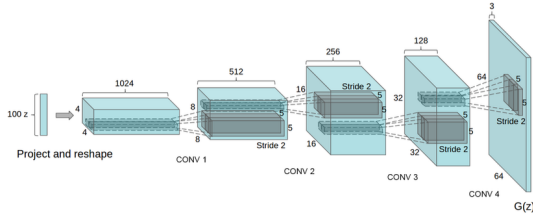
There are many already designed textile prints freely available on the internet, that have been made by expert professionals. The textile design business can benefit from applying machine learning to learn pattern style and, as a result, develop new patterns. However, recent art generation techniques suffer a number of problems and the research is continuously opening new challenges, for e.g., high resolution, greater fidelity in the generated art, diversity in the output images, control over image features, disentangling the desired features, etc.

This paper aims to address the problem of generating new textile patterns by combining machine intelligence into a creative task of generating aesthetically appealing textile designs for local cultures. Our project’s main focus is on locally designed fabric patterns, which will be used to generate new designs following similar style as that of our local prints.

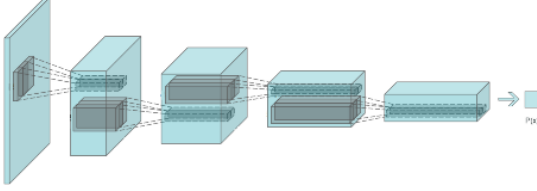
## II. LITERATURE REVIEW

GANs were initially presented in 2014, and since then, they have been a popular tool for state-of-the-art image synthesis. However, early GAN models suffered from model collapse, training instability, and diminishing gradients. Newer models have been proposed in an attempt to address these concerns. Deep convolution networks were introduced in GANs in 2015,

and they have subsequently found widespread use, resulting in sharper pictures [13].



(a) DCGAN Generator



(b) DCGAN Discriminator

Fig. 1: DCGAN Architecture

Wasserstein-GANs (WGANs) were introduced in 2017 and employ the Wasserstein distance as a loss function rather than the standard Jensen-Shannon divergence. WGANs seek to address both model collapse and vanishing gradients [2]. In 2018, Relativistic GANs were introduced. These were based on the idea that as the discriminator's probability of classifying fake samples as real increases, its probability of classifying real data as real should decrease [6]. This idea would improve the quality of the generated samples, and would also stabilize training of the model. As earlier GAN models required hundreds of thousands of images as the training dataset for the model to work, the newer datasets might achieve same or even better results with much fewer data. In 2019, StyleGANs, a style-based architecture that uses an adaptive discriminator augmentation technique to stabilize training while training with restricted data sets, was then developed [7].

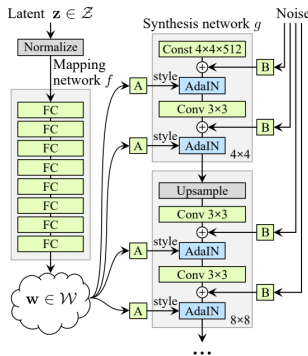


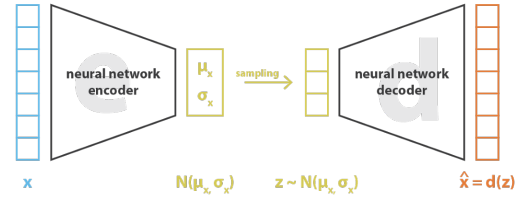
Fig. 2: Style-based Generator Architecture

Attempts have previously been made to classify textile design patterns using GANs [4], where the performance of

image generative models like Wasserstein Generative Adversarial Networks Gradient Penalty (WGANs GP), Deep Convolutional GANs (DCGANs) and Convolutional Variational Autoencoders (CVAEs) for different classes of textile patterns were compared. Moreover, a method of the generation of clothing images for pattern makers using Progressive Growing of GANs (P-GANs) has also been proposed [8]. Some of these techniques have focused on improving the discriminator network e.g., using multiple discriminators, multi-resolution discrimination, or self-attention. On the generator, the focus was mostly on the shape of the input space or the exact distribution in the input space using clustering, Gaussian mixture models, and encouraging convexity in the models.

In another attempt [16], GANs were used for design inspiration. However, these were limited to western-wear and the results generated new t-shirts, jackets and handbags.

Another popular deep learning based generative model that has gained popularity over the year is Variational Autoencoder (VAEs). In a VAE, the input data is sampled from a parametrized distribution, and the encoder and decoder are jointly trained so that the output minimises a reconstruction error in the sense of the Kullback-Leibler (KL) divergence between the parametric posterior and the true posterior.



$$\text{loss} = \|x - \hat{x}\|^2 + \text{KL}[N(\mu_x, \sigma_x), N(0, I)] = \|x - d(z)\|^2 + \text{KL}[N(\mu_x, \sigma_x), N(0, I)]$$

Fig. 3: Variational Autoencoder (VAE)

Not much work has been done to construct local Pakistani fabric pattern designs using GANs, which is what our paper aims to focus on.

### III. DATASET

#### A. Data Collection

We will be using the open source dataset of textile prints [11]. This data set contains about 15K images. Each image is of the size 256 by 256 and belongs to one of six different categories; checkered, dotted, floral, solid, striped, zig zag. These images are taken from actual fabrics, and some of them are of low-fidelity. Therefore, we would need to augment our dataset in order to reach the desired number of prints required, and get the expected output from GANs.

#### B. Data augmentation

To incorporate our locally designed fabric prints and increase the size of our dataset for training, we will collect data through web-scraping from online resources of freely

available Pakistani fabric prints. Some samples from our dataset are shown below.

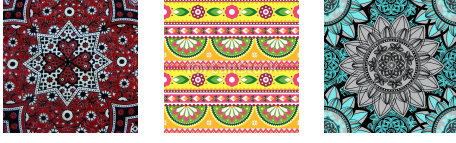


Fig. 4: Fabric prints dataset

#### IV. NOVELTY

Computational creativity has attracted a lot of attention in the research community. The generation of new creative designs is an arduous task which requires a lot of brain work. One of the most important phases in product manufacturing is design generation. However, not much focus has been put on automating the design generation aspect of the textile industry in Pakistan.

All related previous work done in this domain either focused on general fabric patterns or western wear, with no particular focus on our local Pakistani textile industry and its prints. Therefore, to address local fashion industries residing in Pakistan, the aesthetics of our generated patterns will primarily be based on sub continental heritage and designs. In our project, we have mainly focused on Pakistani fabric prints available on the internet by web-scraping websites that are Indian/Pakistani based.

Moreover, for our project we have implemented three different models to check which architecture would give us the most appealing results.

#### V. MODEL & EXPERIMENTS

##### A. Experimental Setup

We compared the performance of DCGAN, StyleGAN and VAEs by using images of size of 64 by 64 and a dataset containing 2K images. Since DCGAN outperformed the two other models, we trained the further trained the DCGAN on a dataset containing 15K images of size 256 by 256.

TABLE I: The parameters used to compare the performance of DCGAN, StyleGAN and VAE.

Parameter	Values
Epochs	1000
Batch Size	24
Number of Image Channels	3
Learning Rate	0.0002
Latent Dimension	512
Number of Images	2K

TABLE II: The value of the hyper parameters used to train the final DCGAN model

Parameter	Values
Epochs	12,000
Batch Size	24
Number of Image Channels	3
Learning Rate	0.00002
Latent Dimension	100
Number of Images	15K

##### B. VAE

We implemented our VAE model consisting of two neural net models; encoder and the decoder. Our encoder consists of five Convolutional layers followed three dense layers. Each Convolutional layer is followed by batch normalisation. The activation function used between each Convolutional layer is ReLU. Our decoder consists of one dense layer followed by Upsample layers. Each Upsample layer is followed by Conv2D and Batch Normalization. The activation between each upsampling layer is ReLU.

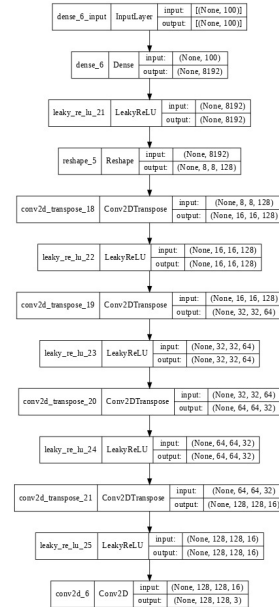


Fig. 5: Architecture of VAE - Decoder

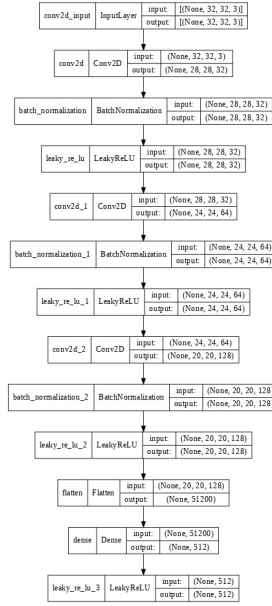


Fig. 6: Architecture of VAE - Encoder

### C. StyleGAN

We have implemented our StyleGAN model the same as the original model that was presented so no custom diagrams have been included.

### D. DCGAN

Each of our DCGAN model consisting of two neural net models; generator and discriminator. Our generator model consists of five Conv2DTranspose layers with one Dense Layer with LeakyRelu activation function between these layers, and tanh function as the activation function for the last layer. Our discriminator consists of six Conv2DTranspose Layer, one Dense Layer with LeakyRelu activation function between these layers, and sigmoid function as the activation function for the last layer.

The structure of the DCGAN which works on 64 by 64 images is similar to the DCGAN which works on 256 by 256 images. The generator now consists of thee Conv2DTranspose layers instead of five and the discriminator consists of four instead of six layers.

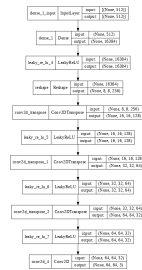


Fig. 7: Architecture of DCGAN model(64 by 64) - Generator

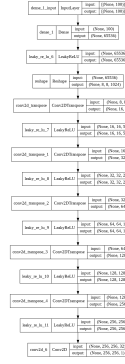


Fig. 8: Architecture of DCGAN model(256 by 256) - Generator

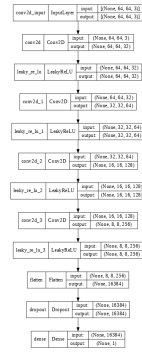


Fig. 9: Architecture of DCGAN model(64 by 64) - Discriminator

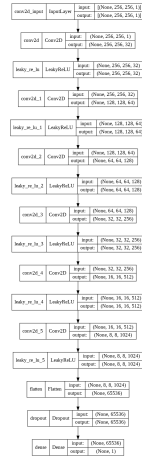


Fig. 10: Architecture of DCGAN model(256 by 256) - Discriminator

## VI. RESULTS & DISCUSSION

### A. VAE

The results that were generated by the VAE were blurry and lacked any distinct shape. However, they contained some form of resemblance to the fabric patterns that were presented in the dataset.

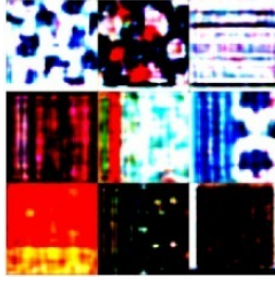


Fig. 11: Results generated by the VAE - 64 by 64 Image

### B. StyleGAN

The results that were generated by the StyleGAN were quite blur and lacked any distinct pattern or shape.

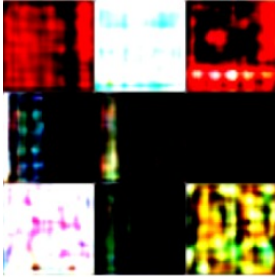


Fig. 12: Results generated by the StyleGAN - 64 by 64 Image

### C. DCGAN

The results that were generated by the DCGAN were also blur but contained less variance and noise as compared to the outputs generated by the VAE and StyleGAN model.

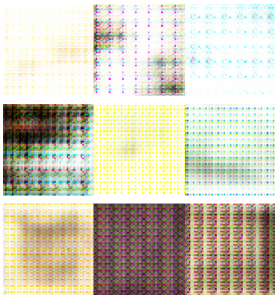


Fig. 13: Results generated by the DCGAN - 64 by 64 Image

### D. Discussion

Our DCGAN model outperformed the StyleGAN and VAE model. Consequently, we trained the DCGAN on a dataset with 15K images of size 256 by 256. The results can be seen in Figure 14.

Our results are an improvement because the previous models generated low-resolution images of size 64 by 64 but our model produces high resolution images of size 256 by 256. Increasing the size of images has also reduced the noise that was found in the previous results. In addition, the results

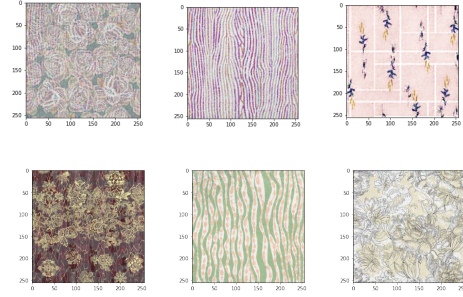


Fig. 14: Results generated by the DCGAN - 256 by 256 Image.

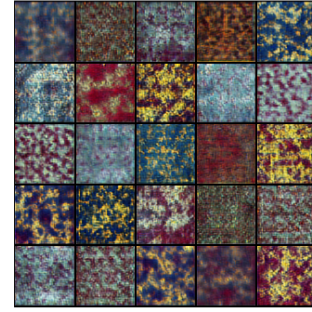


Fig. 15: Previous Results [9]. DCGAN generating image of size 64 by 64 with noise.

produced by our VAE model were similar to the results produced by the previous VAE models.

One explanation for these results is that the DCGAN suffers from over fitting. Consequently, the patterns that are generated are sharp but have a low variance. On the other hand, VAE and StyleGAN models try to learn all the different patterns in the dataset and then combine these patterns to generate a new pattern. However, since the training dataset has a high variance when these different patterns are mixed the resulting image has noise.

## VII. FUTURE WORK

In our project, we focused on an architecture centric approach; we implemented the model that we want to use first and then collected data for it. To get better results, a data centric approach should be taken to solve this problem of automating the process of textile designs, as we need to implement such a model that works best on the type of data available to us.

Moreover, our scrapped dataset has a lot of variance in it. All our selected patterns used to train our model on are each very different from one another. There is little to no similarity between these designs, due to which its harder for our model to learn features. To overcome this problem and get better results, a dataset with fewer variance needs to be collected for such a problem. For example, focusing only on Ajrak prints or Geometric prints would be ideal.

Furthermore, there is no way to measure the aesthetic value of design using mathematical formulas. The evaluation

of a good design is what appeals to the eye, and its not possible that all the designs that our model generates would be aesthetically pleasing. Therefore, working on calculating the aesthetic value for each design, or incorporating human input to select and reject the generated designs could be something to work on in future.

Due to our limitations in this project, we could not train our model for longer periods of time. Training these models for larger periods of time increases the model's ability to learn new features. To make this possible, use of a GPU to train the model would help for future work in this domain.

## REFERENCES

- [1] Zameer Ahmed. Research report on pakistan's textile industry analysis. *SSRN Electronic Journal*, 2021.
- [2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017.
- [3] Haoye Dong, Xiaodan Liang, Bochao Wang, Hanjiang Lai, Jia Zhu, and Jian Yin. Towards multi-pose guided virtual try-on network, 2019.
- [4] Raja Fayyaz, Muaz Mabool, and Muhammad Hanif. Textile design generation using gans. pages 1–5, 08 2020.
- [5] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14*, page 2672–2680. MIT Press, 2014.
- [6] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan, 2018.
- [7] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, 2019.
- [8] Natsumi Kato, Hiroyuki Osone, Kotaro Oomori, Chun Wei Ooi, and Yoichi Ochiai. Gans-based clothes design: Pattern maker is all you need to design clothing. New York, NY, USA, 2019. Association for Computing Machinery.
- [9] Saad Khalil, Muhammad Taha, Rizwan Ali, Hasan Mehmood, and Hammad Dilpazir. Title: Generation of textile patterns through generative adversarial networks generation of textile patterns through generative adversarial networks. 03 2020.
- [10] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network, 2017.
- [11] Alexander J. Medeiros, Lee Stearns, Leah Findlater, Chuan Chen, and Jon E. Froehlich. Recognizing clothing colors and visual textures using a finger-mounted camera: An initial investigation. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility - ASSETS '17*, 2017.
- [12] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. Mirrorgan: Learning text-to-image generation by redescription, 2019.
- [13] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016.
- [14] Amit Raj, Patsorn Sangkloy, Huiwen Chang, Jingwan Lu, Duygu Ceylan, and James Hays. Swapnet: Garment transfer in single view images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.
- [15] Scott Reed, Zeynep Akata, Xinchun Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis, 2016.
- [16] Othman Sbai, Mohamed Elhoseiny, Antoine Bordes, Yann LeCun, and Camille Couprie. Design: Design inspiration from generative networks. *CoRR*, abs/1804.00921, 2018.
- [17] Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár. Amortised map inference for image super-resolution, 2017.