African Motif Pattern Generation using Generative Adversarial Network

Eliel Akinbami - elielkin@stanford.edu
Nnamdi Jacob - nnamdimj@stanford.edu

Abstract

Using Generative Adversarial Networks (GAN), we train a model to generate African motifs. In particular, we train 2 different GAN models and quantify the effect of the number of training steps on the quality of images produced given the small size of our training data.

1. Introduction

African patterns are central to a lot of African cultures. These patterns are made up of complex geometries and color transformations, and can be found on different household materials. Typically, Ankara designs are made through an Indonesian wax-resist dyeing technique called batik[1].

The designs are first created and then imprinted on the fabric. With the advent of technology, printing designs can now be done digitally[2]. This simplifies the design printing phase and reduces cost. This leads to our first motivation, which is to further reduce cost by simplifying the design creation step. Advances in technology have also led to a monopoly on African fabric generation by the European and Chinese markets[3]. Our second motivation is therefore to provide Africans with the tools needed to produce their designs and be able to compete on the global market. This process begins with mass creation of designs using deep learning tools, and eventually a pipeline where design parameters such as shape, color scheme and pattern choices can be inputed and suitable patterns meeting those criteria are generated.

To limit the scope of the project, we focused on a style of fabric pattern called "Ankara"[1]. Here, we propose a methodology for creating these complex colorful patterns using GAN. The result will be a network that generates Ankara patterns that are indistinguishable from batik-made patterns. With this model, it will be possible to create novel Ankara patterns, and further diversify and contribute to African print evolution.

2. Related Work

While, to the best of our knowledge, there has been no work done on African pattern generation, Shen *et al.*[4], and Dibia V[5], have used GANs to create garments and african masks.

In Shen *et al.*[4], the authors train a model to reconstruct garment models that can be adapted to different body shapes. To do this, the training dataset consists of different garments (sewing patterns), human shapes and body motion. A conditional GAN is trained on this dataset to learn a label mask(UV representation of garment) which becomes input to a reconstruction model to create a garment that fits a given body dimension.

In Dibia V[5], a manually curated set of ~9300 african masks were trained using a DCGAN model to generate 64px and 128px images. On visual inspection, the 64px images have more diversity than the 128px images. They further explored the effect of dataset size on generated images.. Dataset lower than 6000 images increased the occurrence of partial mode collapse.

The result from Dibia V. was important as we are training on a small dataset. We decided to use lightweight gan[6,7], a model that has been shown to perform well on small dataset sizes. We also do some comparisons with StyleGAN2[8,9].

3. Data

Our initial dataset included about 15,000 sourced images from pinterest. These images had different sizes, backgrounds, and lighting conditions. Using all the images in this dataset in their initial form as training examples would have distorted the generated images and hampered our goal of generating realistic Ankara patterns. To remedy this, we pre-processed the data and curated a high-quality dataset of 500 images, each of size 256 by 256, without any background. Our goal thus became to assess how well we can train a model to generate Ankara patterns given a small dataset. Both lightweight-gan and StyleGAN2 have an argument, “aug-prob” and “aug-types” that determine the probability of augmenting an image and the types of augmentations possible - translation, cutout and offset. These arguments increase the number of images being fed to the discriminator.

4. Method and Experiment

Given the small size of our training example, we settled on training using lightweight-gan. Lightweight-gan is optimized for minimum computing cost using the following techniques:

- Designing a Skip-Layer channel-wise Excitation module by applying channel wise multiplications in skip-connection as opposed to element-wise addition. This makes it possible to perform skip-connection on layers with different resolutions. Due to space constraint, check out the paper in ref. 8 to see a pictorial representation of the Generator
- Using a self-supervised discriminator D, which is trained as a feature encoder with extra decoders.

Loss: Lightweight-gan originally hinge loss, with the option to use dual contrastive loss, a loss function that has been shown to improve quality over other loss functions[10]. As will be outlined in the results, we start out by training using hinge loss and 1500 training steps. This was our preliminary model to see the quality of generated images. We then train using 15000, 42000 and 92000 steps, toggling between hinge and dual contrastive. Finally, we use StyleGAN2 to train for 37 epochs and compare to models trained using lightweight-gan.

5. Results

To quantitatively analyse generated images, we used the Frechêt Inception Distance (FID) metric[11], which summarizes inception distance from extracted features between real and generated images. Below is a table summarizing the results we get.

Run name	Train steps	Epoch No.	FID	Model	Loss Function
Default-92	92000	92	184	Lightweight-gan	Hinge
GANkara_7	37000	37	188	StyleGAN2	Hinge
GANkara_1	43000	42	176	Lightweight-gan	Dual Contrastive

GANkara_4	15000	15	186	Lightweight-gan	Dual Contrastive
Default-15	15000	15	202	Lightweight-gan	Hinge

Table 1: FID scores for different training steps and loss functions

Qualitatively, the first run with 1500 steps generates very noisy images. This is understandable as this is from 1 epoch. At 15 epochs, we begin to see the emergence of patterns (fig. 1 in appendix) similar to real life patterns. However, these images are also riddled with noise. At 92 epochs, the images begin to look more realistic(fig. 2 in appendix), however, with noise. Another observation is that at lower epochs, the trained model is better able to generate images with different blends of color and shapes. Visually, the model does poorly in generating images with color on the grayscale. This makes sense for 2 reasons:

1. Most of our dataset contains images that are rich in color, as is the case with most ankara fabric.
2. Given the limited size of our training example, it is easier for the model to learn and get features from images with more information. This makes it easier for images rich in color to pass the generator-discriminator game, while images with grayscale may have a higher failure probability.

6. Conclusion and Future work

Using a training example = 500 images, and lightweight-gan and StyleGAN2 to train, we set out to generate realistic ankara fabric and ended up adjusting our goal to evaluating lightweight-gan with different training sizes and loss function. The quantitative metric - FID - shows an 8% decrease when loss function is changed from Hinge to Dual Contrastive, which is consistent with the Heusel *et al.*[7,9,11]. However, there is no significant difference in FID between epoch 92 and 42. From visual inspections, there is an improvement in quality of generated images from epoch 1 to epoch 92. This means our results are inconclusive. However, this informs possible future steps for this work.

1. To get smoother generated images, use training steps > 100k
2. Keep using dual contrastive loss as the loss function
3. Work on processing data available to meet criteria - no background, 256x256
4. Train model using supervised learning so as to be able to specify image color, shape etc.

References

1. Oyairo, M. (2020, February 1). *A brief history of Ankara fabric and African prints*. Oludan. Retrieved October 2021, from https://www.oludan.com/blogs/news/a-brief-history-of-ankara-fabric-and-african-prints.
2. *Vlisco wax print - wax printing process / fabric printing ...* VLISCO. (2017). Retrieved December 2, 2021, from https://www.vlisco.com/world-of-vlisco/craft/printing-process/
3. HERSEY, F. (2019, June 4). *Putting the 'African' back in west african wax print fabrics*. France 24. Retrieved December 2, 2021, from https://www.france24.com/en/20190603-africa-wax-print-fabric-textile-pagne-ivory-coast-chinese.
4. Shen, Y., Liang, J., & Lin, M. C. (2020). GAN-based garment generation using sewing pattern images. *Computer Vision – ECCV 2020*, 225–247. https://doi.org/10.1007/978-3-030-58523-5_14
5. Dibia, V. (2019, January 9). *Art + AI-generating African masks using (tensorflow and tpus)*. Medium. Retrieved October 2021, from https://towardsdatascience.com/african-masks-gans-tpu-9a6b0cf3105c.
6. Liu, B., Zhu, Y., Song, K., & Elgammal, A. (2021, January 12). *Towards faster and stabilized gan training for high-fidelity few-shot image synthesis*. arXiv.org. Retrieved December 2, 2021, from https://arxiv.org/abs/2101.04775v1.
7. Lucidrains. (n.d.). *Lucidrains/Lightweight-Gan: Implementation of 'Lightweight' gan, proposed in ICLR 2021, in Pytorch. high resolution image generations that can be trained within a day or two*. GitHub. Retrieved December 2, 2021, from https://github.com/lucidrains/lightweight-gan.
8. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020, March 23). *Analyzing and improving the image quality of stylegan*. arXiv.org. Retrieved December 2, 2021, from https://arxiv.org/abs/1912.04958.
9. Lucidrains. (n.d.). *Lucidrains/stylegan2-pytorch: Simplest working implementation of stylegan2, state of the art generative adversarial network, in Pytorch. enabling everyone to experience disentanglement*. GitHub. Retrieved December 2, 2021, from https://github.com/lucidrains/stylegan2-pytorch.
10. Yu, N., Liu, G., Dundar, A., Tao, A., Catanzaro, B., Davis, L., & Fritz, M. (2021, October 7). *Dual contrastive loss and attention for Gans*. arXiv.org. Retrieved December 3, 2021, from https://arxiv.org/abs/2103.16748v2.
11. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). *Gans trained by a two time-scale update rule ... - nips*. Retrieved December 3, 2021, from https://papers.nips.cc/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf.

Appendix

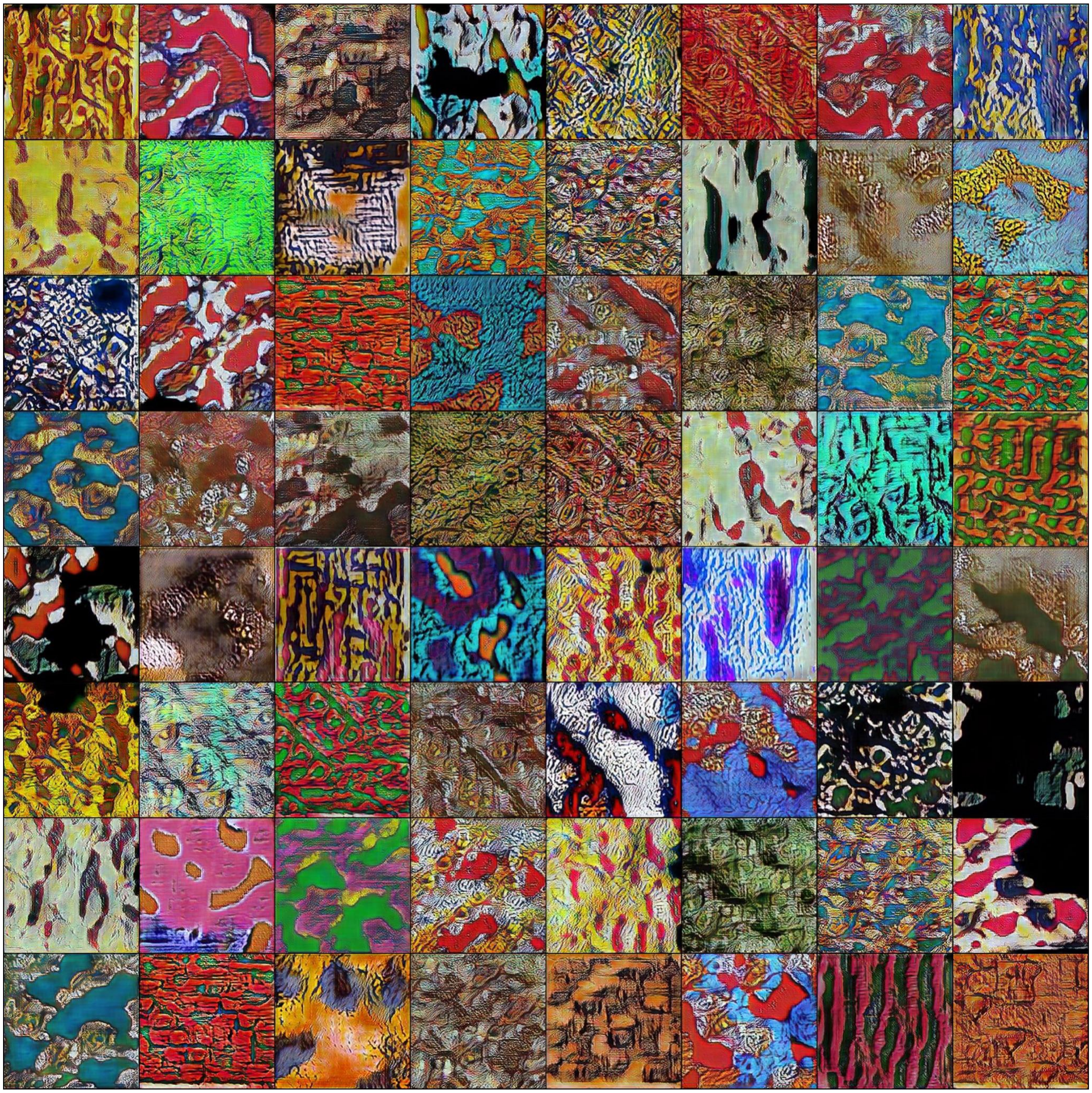

Fig 1: Sample images generated after 15 epochs. Images that have a lot of color are more defined that images with colors that fall on the grayscale

Fig 2: Sample images generated after 92 epochs. The first 10 images show samples that look similar to real ankara images while the last 5 show instances of possible mode collapse

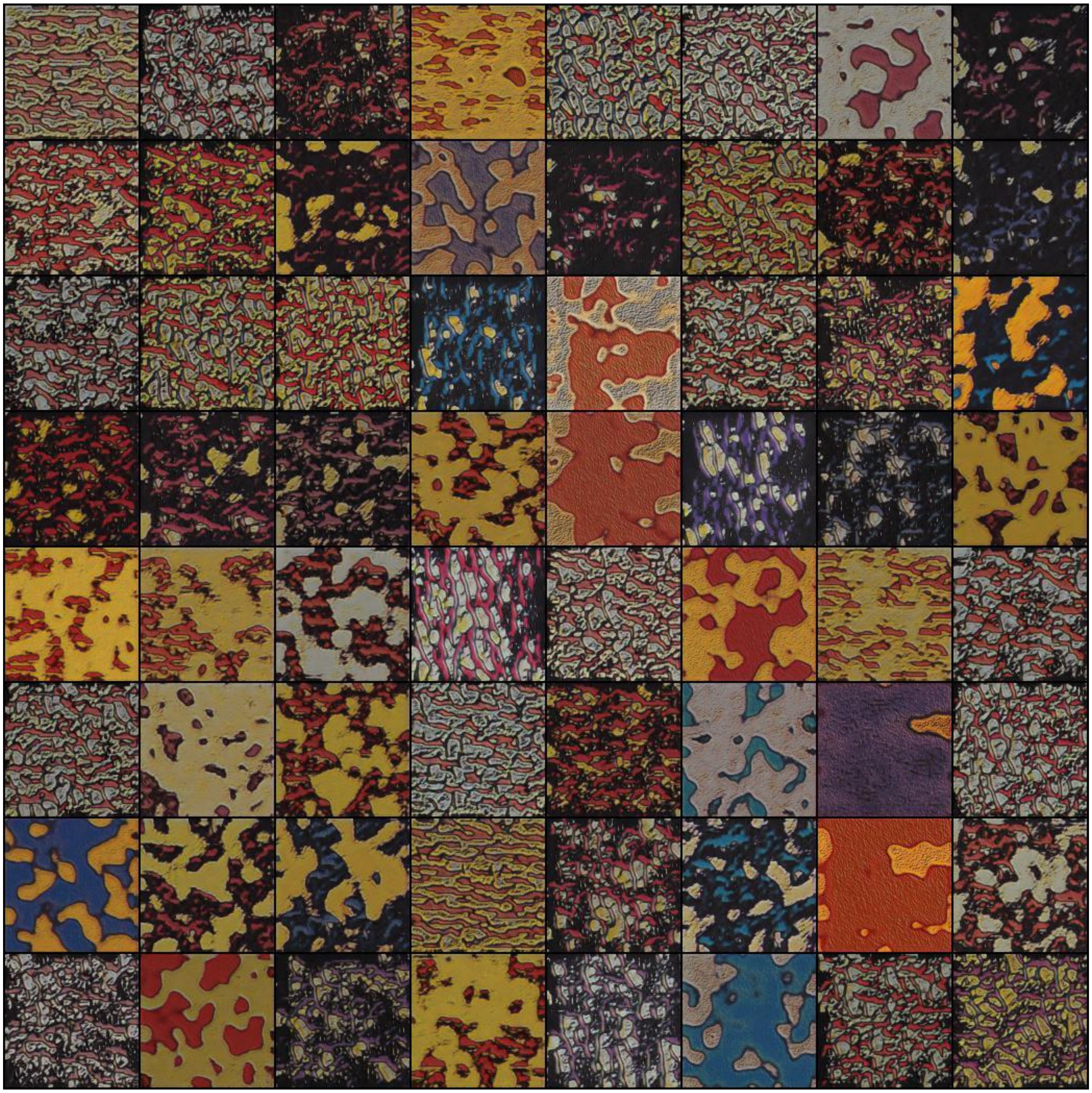

Fig 3: Images generated after 37 epochs using StyleGAN2. While these images seem to have less noise, they are not as diverse (color and shape wise) when compared with images generated by lightweight-gan.