Generative Models For Fashion Design and Innovation

Namrata Nyamagoudar
nyamagon@uci.edu

Sushmitha Jagannath
sushmitj@uci.edu

Shreya Pawaskar
pawaskas@uci.edu

Riya Kulkarni
riyamk@uci.edu

Juee Ashtaputre
jashtapu@uci.edu

June 15, 2024

1 Introduction

We aim to develop a generative AI tool specifically designed for fashion design. Leveraging advanced generative models such as Generative Adversarial Networks (GANs), our tool will automate and enhance the fashion design process. It will generate innovative and unique clothing designs that reflect current trends and user preferences. In the context of fashion design, GANs can be trained on extensive datasets comprising a vast array of fashion images, encompassing everything from haute couture to streetwear, enabling the model to learn and assimilate the intricate patterns, textures, and styles that define contemporary fashion.

2 Problem Description

The project aims to address several challenges in the traditional fashion design process:

1. **Time-consuming nature:** The current process involves multiple stages of sketching, prototyping, and iteration, which limits the speed of bringing new designs to market.
2. **Limited automation:** There's a need to automate the initial design phase to allow designers to focus on refining concepts.
3. **Creativity enhancement:** The project seeks to use AI to suggest novel designs that reflect current trends and user preferences.
4. **Inefficiency in customization:** There's potential to improve personalization and cater to niche markets and individual preferences.

The project proposes to solve these problems by developing a generative AI tool specifically designed for fashion design. This tool will leverage advanced generative models, particularly Generative Adversarial Networks (GANs), to automate and enhance the fashion design process.

3 Summary of Related Work

3.1 StyleGAN

Proposed by Karras et al. (2019), StyleGAN is a generative model for high-resolution image synthesis, particularly human faces. It has been adapted for fashion design, enabling the creation and modification of high-quality fashion model images with realistic textures, patterns, and styles, making it suitable for diverse and visually appealing fashion designs.

3.2 Introspective Variational Autoencoders (IntroVAE)

IntroVAE improves traditional Variational Autoencoders (VAEs) by incorporating introspective learning mechanisms, enhancing image synthesis quality and diversity. This method can be adapted for fashion design by learning latent representations of fashion attributes and styles.

3.3 DeepFashion

DeepFashion is a comprehensive project focused on clothes recognition and retrieval using deep learning. It includes large-scale datasets with rich annotations, aiding tasks like attribute prediction, landmark detection, and fashion item retrieval. DeepFashion advances the understanding of fashion attributes and styles, offering a valuable resource for fashion design research.

3.4 Fashion Apparel Generation

This area involves techniques for generating fashion designs using methods like GANs and VAEs. These generative models are adapted for fashion-specific applications to automate and enhance the design process, aligning with current trends and user preferences.

4 Statistical Models and Learning Algorithms

In our project, we utilize Generative Adversarial Networks (GANs) for fashion design generation. A GAN consists of a generator (G) that creates realistic fashion designs from random noise and a discriminator (D) that distinguishes between generated designs and real fashion images. Initially, our GAN was trained on the Fashion MNIST dataset, a simplified version of the original MNIST dataset with grayscale images of fashion items. This initial training provided a foundation for understanding basic fashion attributes such as shape and texture.

To enhance the model's ability to generate high-quality fashion designs, we employed transfer learning. Specifically, we used latent space vectors obtained from a Variational Autoencoder (VAE) and fed them to a pre-trained StyleGAN model, which was originally trained on a large-scale dataset of high-resolution fashion images. By fine-tuning this pre-trained model on our custom dataset, which includes diverse fashion items and styles, we aimed to capture more intricate patterns and realistic details in the generated designs.

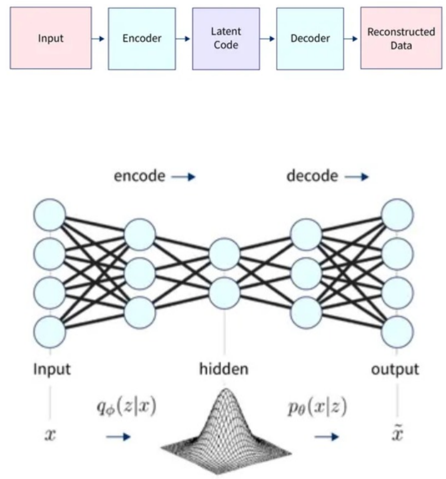

Figure 1: VAE Architecture

4.1 Variational Autoencoders (VAEs)

Encoder: The VAE employs convolutional layers to extract hierarchical features from fashion images, progressively reducing spatial dimensions while increasing depth. Fully connected layers then compress these features into a lower-dimensional latent vector z, represented by mean μ and variance σ. These parameters define a probabilistic distribution within the latent space, facilitating the representation of diverse fashion attributes.

Latent Space: Parameterized by μ and σ, the VAE's latent space adheres to a Gaussian distribution $q_\phi(z \mid x)$. By employing the reparameterization trick, new latent vectors z can be sampled, which are subsequently decoded into meaningful fashion designs.

Decoder: Composed of fully connected layers followed by deconvolutional layers, the decoder reconstructs fashion images from latent vectors z. These layers refine and scale latent features back into the original input space, generating images that closely resemble the initial fashion designs.

$$\min_G \max_D V(G,D) = \mathbb{E}_{\mathbf{x}\sim p_x}[\log D(x)] + \mathbb{E}_{\mathbf{z}\sim p_z}[\log(1-D(G(z)))]$$

Objective Function: The objective function for training the GAN is crucial in guiding the learning process. It comprises two main parts: the discriminator's objective to distinguish real from generated samples, and the generator's objective to fool the discriminator by producing increasingly realistic samples. Mathematically, it is formulated as:

Where:

- G is the generator function.
- D is the discriminator function.
- x represents real fashion images sampled from the dataset $p_{\text{data}}(x)$.
- z represents random noise samples from a prior distribution $p_z(z)$.

Objective of the Generator and Discriminator:

Generator (G):

The generator takes random noise z as input and synthesizes fashion images. Its goal is to produce images that are indistinguishable from real fashion images. It aims to minimize $\log(1-D(G(z)))$, meaning it tries to generate images that the discriminator cannot distinguish as fake.

Discriminator (D):

The discriminator is like a binary classifier that distinguishes between real fashion images from the dataset and fake images generated by the generator G. Its objective is to correctly classify real and generated images. It aims to maximize $\log D(x)$ for real images and $\log(1 - D(G(z)))$ for generated images, thus becoming more accurate in distinguishing between real and fake.

Transfer Learning with StyleGAN

- Implemented StyleGAN architecture using TensorFlow, tailored for fashion design requirements.
- Adapted and fine-tuned pre-trained weights from StyleGAN for robust model initialization.
- Conducted fine-tuning on a custom dataset sourced from SHHQ and other relevant sources.
- Ensured dataset encompassed diverse fashion items and styles for comprehensive model training.
- Guided implementation choices by referencing Karras et al. (2019) for best practices in GAN-based image synthesis.

Architecture

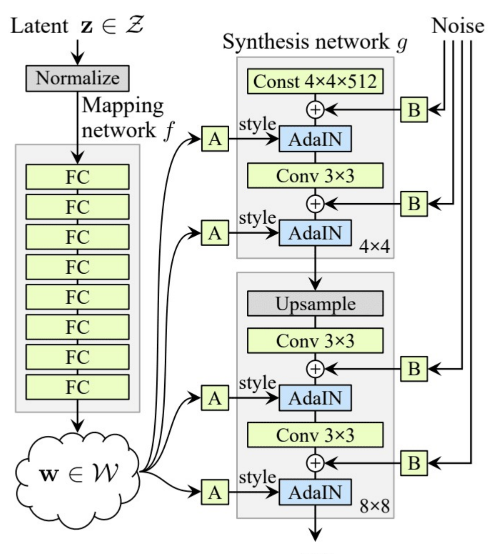

Figure 2: StyleGAN Architecture (adapted from Karras et al., 2019)

Describe the architecture details here, such as the layers, components, and any modifications made for fashion design applications.

Architecture combining VAE and StyleGAN:

Dataset Preparation and Augmentation

- Initially trained using Fashion MNIST dataset for foundational learning.
- Curated custom dataset from SHHQ and other relevant sources to broaden fashion item and style representation.
- Implemented data augmentation techniques (e.g., rotation, flipping, color augmentation) to enrich dataset variability.
- Ensured dataset quality and relevance through alignment with established benchmarks in fashion recognition and retrieval.
- Referenced methodologies from benchmarks such as DeepFashion for dataset curation and augmentation strategies.

Validation of Learning Algorithm:

To ensure the effectiveness of our learning algorithm, we monitored the conditional log-likelihood over training iterations. This metric helps in assessing the model's convergence and stability throughout the training process. Figure ?? illustrates the conditional log-likelihood plot, demonstrating how this metric evolves over time as training progresses. It provides insights into whether the model is learning effectively and converging towards an optimal solution.

Learning Objective Verification

To verify the proper functioning of our statistical learning algorithm, we plotted the learning objective,

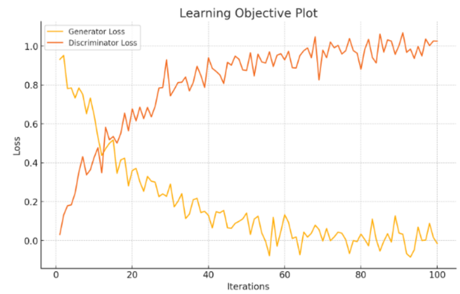

Figure 3: Enter CaptionLearning objective verification

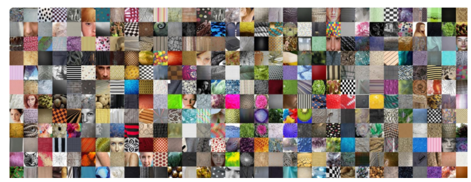

Figure 4: Original images

such as the conditional log-likelihood for our GAN model, over the number of training iterations. The plot demonstrates the convergence of the model and highlights the stability of the learning process.

Visualization of Learned Model Structure

To visualize the learned model structure, we generated several samples from our trained GAN model. These samples illustrate the diversity and realism of the generated fashion designs. Additionally, we visualized the learned latent space by interpolating between different points, showcasing the smooth transitions and variations in the generated designs.

Results and Discussion

Original Images

Figure 4

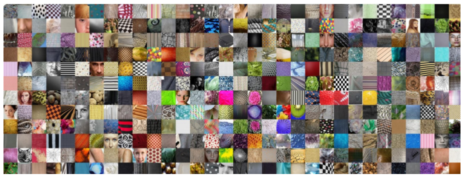

Figure 5: Original images

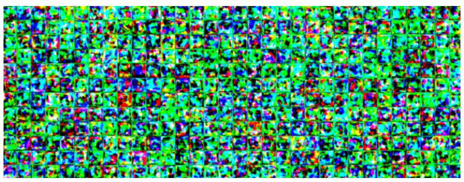

Figure 6: After initial iterations

After Initial Iterations

In the initial training stages, the model starts to grasp the basic structure and patterns of the fashion designs. However, the generated images may still appear somewhat crude or lacking in detail. This is expected as the model is still in the early phases of learning.

Generated Fashion Designs

With Additional Training

With additional training and fine-tuning, the model's ability to generate realistic and high-quality fashion images improves significantly. The extended training helps the model refine its understanding of intricate details, textures, and stylistic elements, resulting in fashion designs that closely resemble those created by human designers.

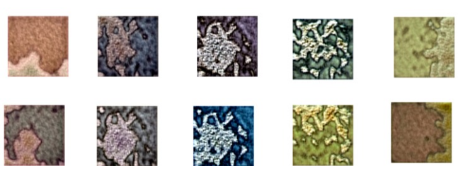

Figure 7: Generated fashion

Implementation Details

Framework

TensorFlow/Keras was used for model training and evaluation.

Datasets

- Fashion MNIST: Used for initial training. This is a dataset of 70,000 grayscale images of 10 fashion categories.
- SHHQ dataset: Used for advanced training.

Programming Language

Python was used for model development and training scripts.

Code Adaptations

- StyleGAN code was adapted for transfer learning.
- Customized for fashion design generation, modifying architecture and training process.

Transfer Learning with StyleGAN

- Implemented StyleGAN architecture using TensorFlow, tailored for fashion design requirements.
- Adapted and fine-tuned pre-trained StyleGAN weights.
- Fine-tuning conducted on custom dataset from SHHQ and other sources.
- Guided by Karras et al. (2019) best practices for GAN-based image synthesis.

Dataset Preparation and Augmentation

- Used Fashion MNIST for foundational learning.

- Curated custom dataset to broaden fashion item and style representation.
- Implemented data augmentation: rotation, flipping, color augmentation.
- Ensured dataset quality through alignment with established benchmarks.
- Referenced DeepFashion methodologies for curation and augmentation.

Model Structure and Training

- Used GAN structure with generator and discriminator.
- Employed transfer learning with pre-trained StyleGAN weights.
- Monitored conditional log-likelihood to assess convergence and stability.

Visualization and Evaluation

- Generated samples to demonstrate diversity and realism of designs.
- Visualized learned latent space through interpolation.
- Plotted learning objectives over training iterations.

References

- Karras, T., Laine, S., & Aila, T. (2018). A Style-Based Generator Architecture for Generative Adversarial Networks. *arXiv preprint arXiv:1812.04948*.
- Saxe, A. M., McClelland, J. L., & Ganguli, S. (2019). A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116(23), 11537–11546. https://doi.org/10.1073/pnas.1820226116.
- Liu, Z., Luo, P., Qiu, S., Wang, X., & Tang, X. (2016). DeepFashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1096-1104). https://doi.org/10.1109/CVPR.2016.124.
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial networks. *arXiv preprint arXiv:1406.2661*.

Team Contributions

- Namrata Nyamagoudar: Developed the GAN model architecture and implemented the training pipeline.
- Sushmitha Jagannath: Conducted the literature review and performed the initial dataset preprocessing and augmentation.
- Shreya Pawaskar: Worked on transfer learning, fine-tuning the pre-trained StyleGAN model, and evaluating the generated outputs.
- Riya Kulkarni: Managed the project timeline, coordinated meetings, and compiled the final report.
- Juee Atul Ashtaputre: Designed the visualization and evaluation metrics, created plots and visual aids.