

FashionGAN

Rishab Kapur, Srijan Singh, Arshit Arora, Yash Nigam

Introduction

The online fashion industry has seen a huge spike in consumers since the pandemic hit the world in 2020. From large brands to small start-ups, the trend of online shopping gained massive popularity. This resulted in designers pushing their limits to produce large amounts of new designs in an incredibly quick time. To tackle this strain, FashionGAN has been introduced. It is a generative adversarial network that produces customised dress designs based on user input. FashionGAN app works on a GAN modifier and searches for products and models. It provides a newly generated image based on users' inputs. It includes features to modify dress size, patterns, sleeve lengths, etc. The scope is to generate base designs for dresses, so that designers only need to work on refinements and details.

Approach

FashionGAN uses a GAN model to implement machine learning and deep learning techniques that feeds on a huge dataset of women's dresses. It uses the Pillow library of Python, with CycleGAN and StarGAN classes to utilize GAN processes.

The usage flow is as follows:

- App starts with an input image
- A shape attribute is selected to modify the shape of the dress. This step also has a skip option to continue with pattern.
- Next, a pattern attribute is selected to modify the type of pattern that is desired on the dress. Currently, it has two options, floral pattern and stripes pattern.
- The model generates an output image with the features that were provided by user as input.
- Select an image from the returned results to start again
- The model search results are only illustrative, so a "continue modification" option is only available when a product image is selected.
- The continue modification option starts the app cycle again with the selected image.

Technology used

We have used a GAN model which works with mainly two classes, CycleGAN and StarGAN.

StarGAN is used for learning mappings among multiple dress images.

CycleGAN, is used for training a deep convolutional neural network for image-to-image translation tasks. The network learns mapping between input and output images using unpaired dataset.

We have used Akiwi50 and RESNET models for obtaining similar dress images and for shape detection of dresses based on size, sleeve length, etc.

We use Pillow library for inputting images into the model.

The code has been written using PyTorch library for executing deep learning techniques.

Finally, we have applied Gradio for UI implementation of the app.

Results

We were able to generate a customised image on the basis of the options/features that we inserted for input. We were also successful in finding similar images in the dataset from the new image we generate.

UI Implementation

For the UI part of this model, we have used the gradio library in python.

Gradio is the fastest way to demo your machine learning model with a friendly web interface

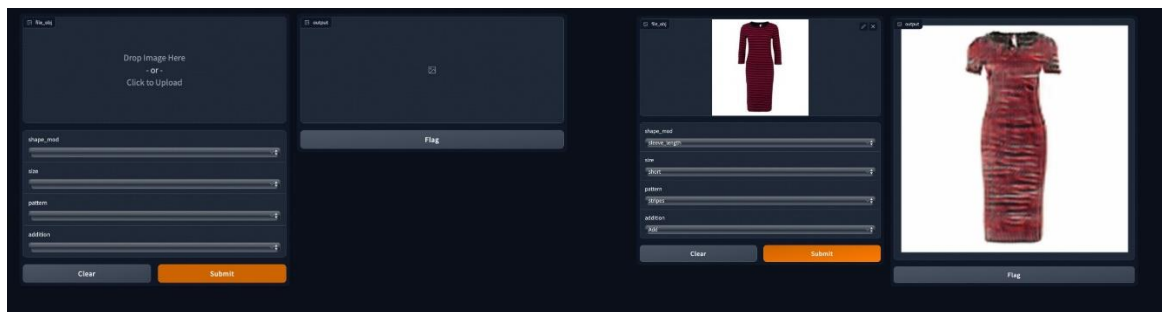
Interface is Gradio's main high-level class, and allows you to create a web-based GUI / demo around a machine learning model (or any Python function) in a few lines of code. You must specify three parameters: (1) the function to create a GUI for (2) the desired input components and (3) the desired output components. Additional parameters can be used to control the appearance and behaviour of the demo.

To use gradio in our system we use the command of :-

```
import gradio as gr
```

Class method that constructs an Interface from a Hugging Face transformers. Pipeline object. The input and output components are automatically determined from the pipeline.

Here are some snapshots of the UI of the model :-



References

1. Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, Jaegul Choo, “*StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation*”, arXiv:1711.09020
2. Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros, “*Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*”, arXiv:1703.10593
3. DeepStyle (Part 2): The Fashion GAN, <https://towardsdatascience.com/deepstyle-part-2-4ca2ae822ba0>
4. ClothingGAN, <https://devpost.com/software/clothinggan>
5. Using GAN for Fashion, <https://medium.com/analytics-vidhya/using-gan-for-fashion-6dc11012ac6a>
6. Qiang Wu, Baixue Zhu, Binbin Yong, Yongqiang Wei, Xuetao Jiang, Rui Zhou & Qingguo Zhou, “*ClothGAN: generation of fashionable Dunhuang clothes using generative adversarial networks*”, <https://doi.org/10.1080/09540091.2020.1822780>
7. Tero Karras, Samuli Laine, Timo Aila, “*A Style-Based Generator Architecture for Generative Adversarial Networks*”, arXiv:1812.04948