AI FASHION

TEAM: Aaliyah Hänni, Dwight Sablan, Liem Luong, Vanessa Hsu
SPONSOR: Tatiana Teppoeva | Microsoft Sr. Data & Applied Scientist

INTRODUCTION

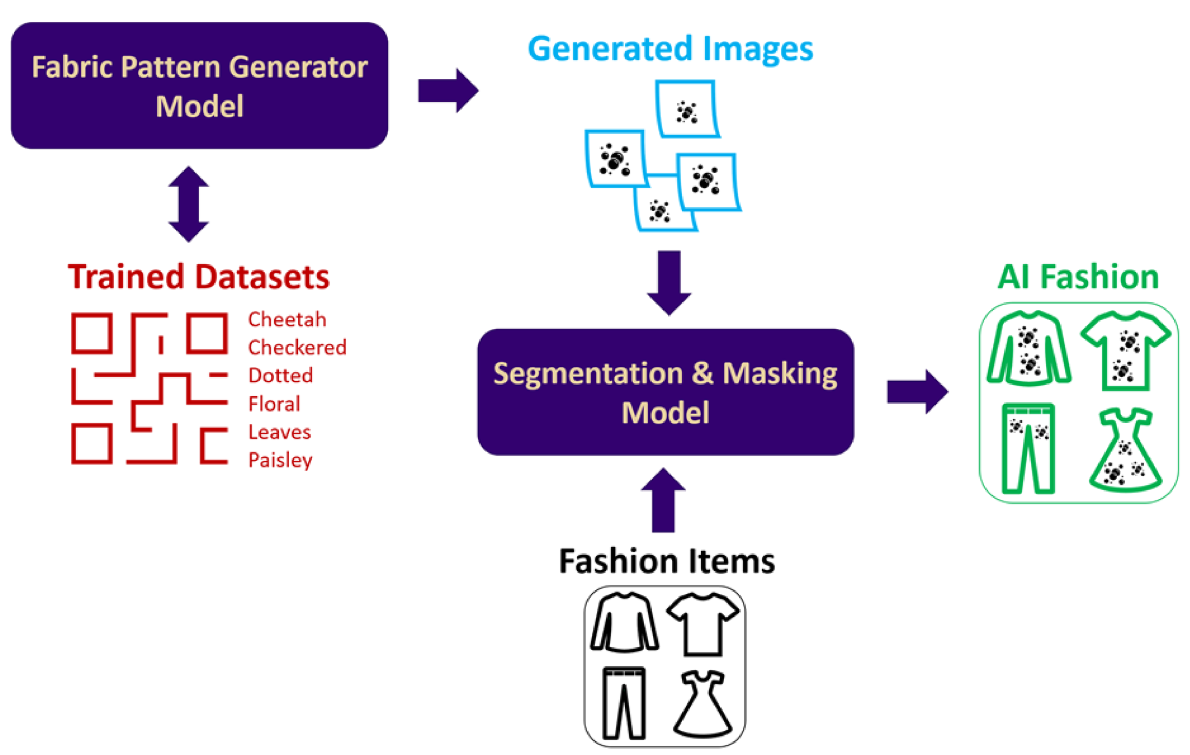
Art and data science, the two fields gain more traction in recent years. The application between ART and AI are the inspiration for people to produce new pieces of art with more accessible to the broader audience.

In the fashion industry there is a constant demand for new and interesting design patterns, which can often be expensive and time consuming to create. Fashion designers are bound by their own internal biases and have limitations on the level of creativity and in-demand fashion that they can create within a given timeframe.

With advance in deep learning, neural networks can be used to alleviate some problems in this field by improving the level of creative in design for both quantity and quality outputs. In this works, we implement the neural network model to generate new textile pattern for many fashion items. Our solutions and findings could be easily used by non-tech savvy and will make design creation more accessible for the broader audience.

Demo: tny.im/CP0Ke

ARCHITECTURE



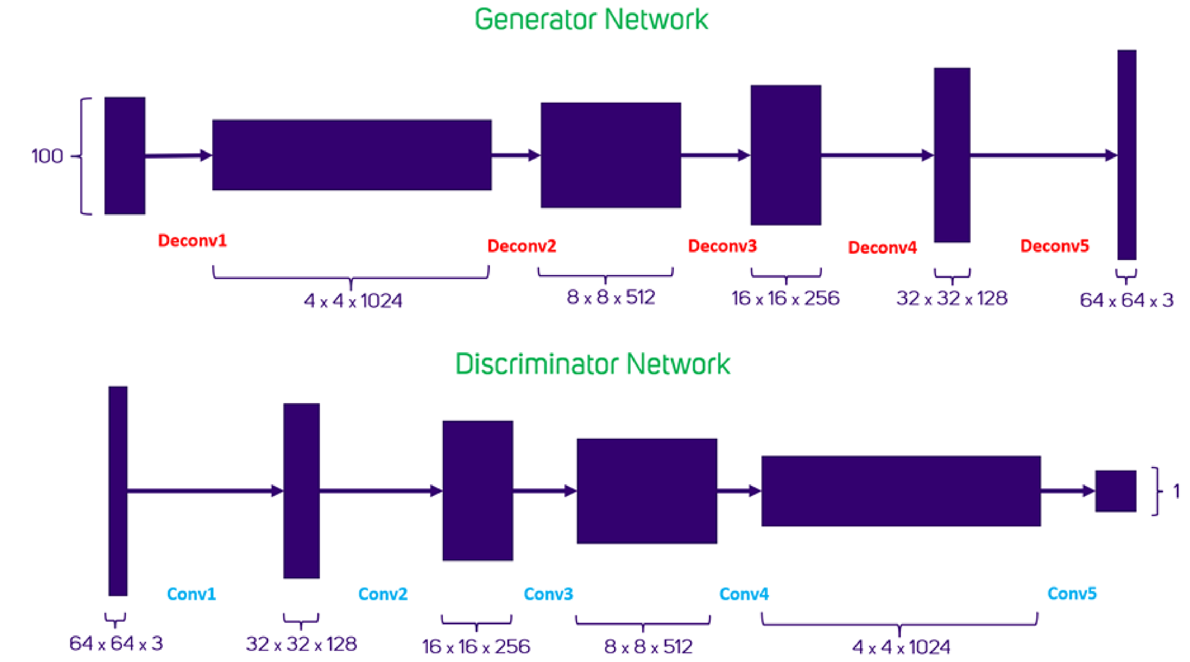
ARCHITECTURE DIAGRAM OF AI FASHION APPLICATION

Our core architecture design includes two main components: Fabric Pattern Generator (component 1) and Segmentation & Masking (component 2).

In component 1, we apply the Deep Convolutional Generative Adversarial Network (DCGAN) by training with various textile pattern datasets: cheetah, checkered, floral, dotted, leaves, and paisley. The output of this component 1 will be a neural network generated image input for component 2. In the component 2, another pre-trained neural network model to perform image segmentation of clothe pattern and masking the new pattern over the fashion items to produce a result.

MODEL

DCGAN Model is the choice being used for fabric pattern generator model in this project. The advantage of DCGAN model is its continuous improving between Generator Network and Discriminator Network on improving the performance while reducing the loss of both networks to produce the best fake images. Below is the neural network structure of the Generator Network and Discriminator Network



DCGAN MODEL

Improving the model performance and reducing the loss are our objectives. We use the default Adam optimizer and Binary Cross Entropy loss from PyTorch.

Loss Functions and Optimizers

$$\ell(x,y) = L = \{l_1,\dots,l_N\}^\top, \quad l_n = -\left[y_n \cdot \log x_n + (1-y_n)\cdot \log(1-x_n)\right]$$

Above is the Binary Cross Entropy loss function for both *log(D(x))* and *log(1 - D(G(z)))*

- Objective of training the discriminator: **maximize the *log(D(x)) + log(1 – D(G(z)))*** with the aim to increase the probability of correctly classifying of input image as real or fake images.
- Objective of training the generator: **minimize the *log(1 – D(G(z)))*** with the aim to generate better fake image outputs.
- *D(x)* is the average output (across the batch) of the discriminator for the all-real batch. In theory, this number should start close to 1 and then converge to 0.5 when G gets better
- *D(G(z))* is the average discriminator outputs for the all-fake batch. In theory, this number should start near 0 and converge to 0.5 as G gets better

With the component 2 (Segmentation & Masking), we leverage the pre-trained model (Unet_2020-10-30). This model weights for clothe segmentation were trained over the Kaggle dataset - iMaterialist (Fashion) 2019 at FGVC6.

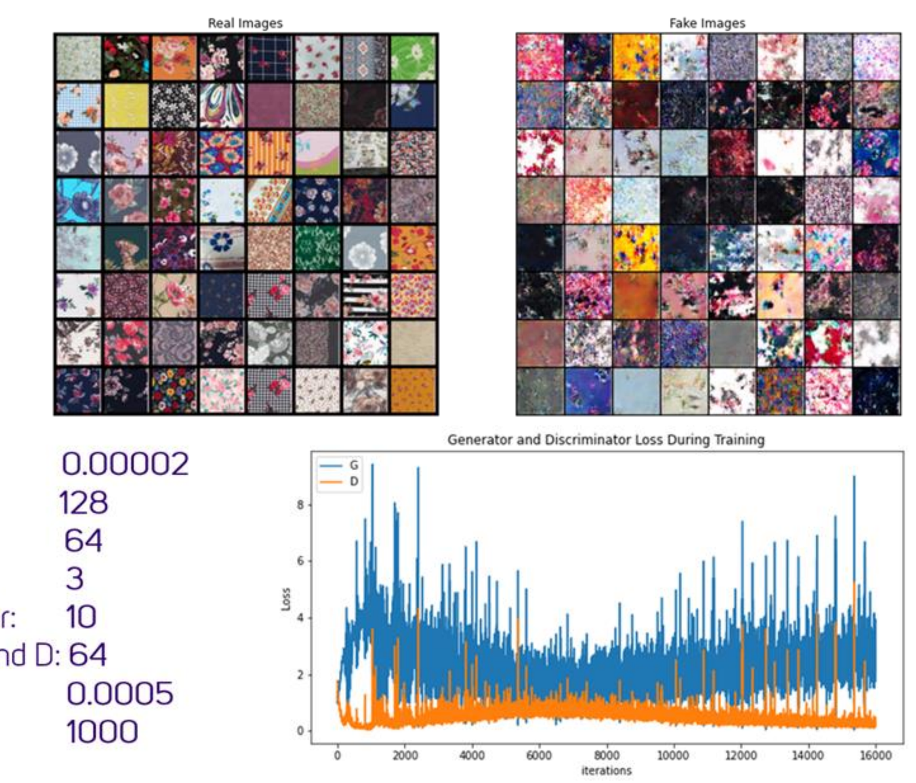
DATA

The training datasets for the DCGAN model are segregated into one of the 11 categories. These images are cleaned and standardized to the same height and width (224 x 224 pixels)

RESULTS

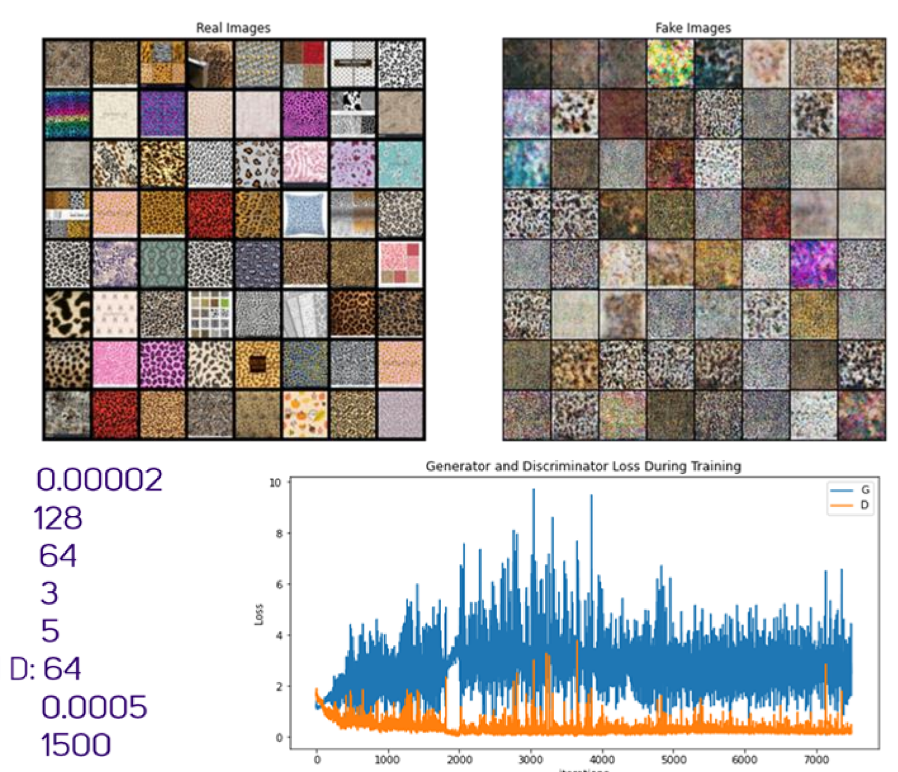
Fabric Pattern Generator Model: Among the six-pattern datasets we trained, we achieved good results for Floral and Cheetah of the fake images and lower loss values

Floral



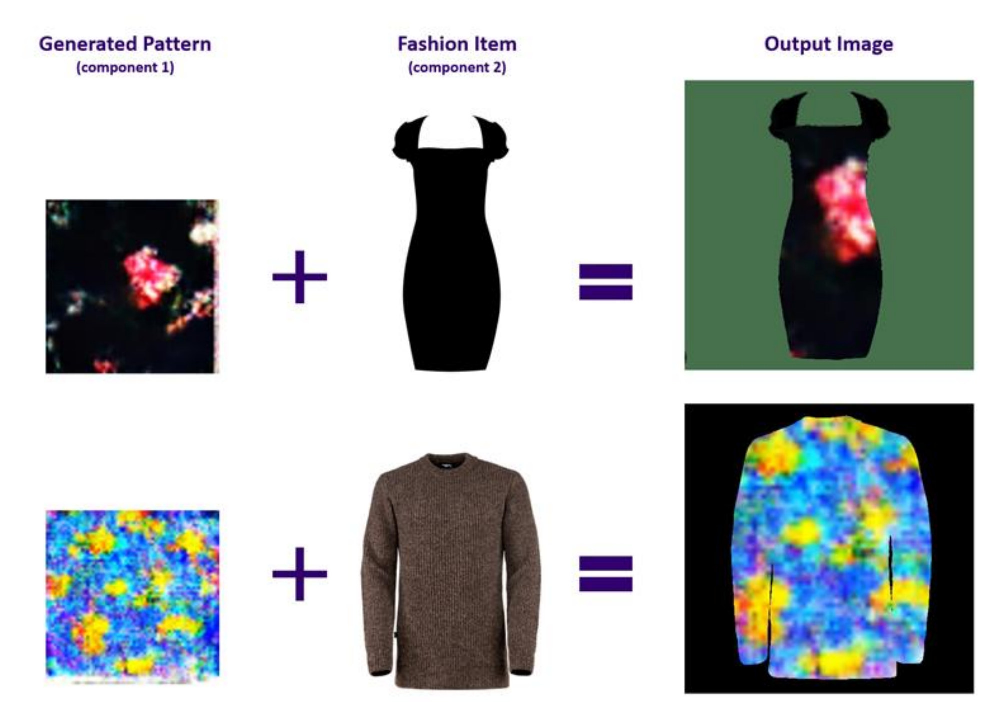
Learning rate: 0.00002
Batch Size: 128
Image Size: 64
Number of Channel: 3
Size of z latent vector: 10
Feature maps in G and D: 64
Adam Optimizer: 0.0005
Epoch: 1000

Cheetah



Learning rate: 0.00002
Batch Size: 128
Image Size: 64
Number of Channel: 3
Size of z latent vector: 5
Feature maps in G and D: 64
Adam Optimizer: 0.0005
Epoch: 1500

Below are sample of the output results after masking generated pattern (component 1) on to a fashion item (component 2)



CONCLUSIONS

Among the six categories, we achieve best results – where the fake images generated from the neural network are closer to the real images - with floral, cheetah, paisley, and leaves. The checkered and dotted are showing some improvements in the fake images but still not the optimum outcome. The take-away from the result is that DCGAN model works well for the type of images that does not possess a strict geometric pattern.

The other interesting observation that we see that can improve the model by training with various settings. This will reduce the loss performance between Generator Network and Discriminator Network :

- A small Adam optimizer value (< 0.05)
- Smaller size of z latent vector (< 100)
- Learning rate between 0.0002 and 0.00002
- The higher training epoch, the better result it yield

Art image can be viewed as an abstract art. This is due to the nature of there is no boundary of how art image is produced and perceived by viewer. Art can have bias as well. A regular viewer may not find the Picasso painting interesting, but a real artist can see the abstract inner beauty of that piece of art. This characteristic is also applied in the neural network generated image.

LIMITATIONS

One of the limitations we experience is the nature of computer vision. This area in deep learning requires computing power – especially GPU – in order to train the neural network model. The better result we gain but it costs lot of time to train.

Second limitation is the limit of free available GPU resources from Google Colab we have.

Third limitation is the accuracy of result from component 2 based on various factors: image characteristic (transparency background, plain icon), blend image with people from obscure view,

FUTURE WORKS

We would love to improve the model performance by using a better-quality datasets as well as updating the neural network layers to improve the accuracy of fake images.

Improving the performance of component 2 so that it can mask the graphic well for various scenarios from the above limitations.

REFERENCES

The following resources we use and refer to when working on this project:

- https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html
- https://github.com/ternaus/cloths_segmentation