DEVELOPMENT OF FASHION CONTENT USING GENERATIVE ADVERSARIAL NETWORKS (GANs) AND PREDICTING THEIR PRICES

by

Echezona .C. Uzoegbo

202311253

A Dissertation

Submitted to the Department of Data Science, Artificial Intelligence and Modelling (DAIM)

Faculty of Science and Engineering

University of Hull

In fulfillment of the requirements

For the degree of the Master of Artificial Intelligence and Data Science

December 2023

ABSTRACT

This study investigates the use of Generative Adversarial Networks (GANs) in the fashion sector, with a particular emphasis on the creation of designs for men's sneakers and the ensuing estimation of their costs. To produce pleasing and realistic images of sneakers, the study makes use of a Deep Convolutional GAN (DCGAN), considering elements like design evolution over epochs and optimization processes to improve model performance.

Two distinct datasets, gathered from Kaggle and Data.World were used for image synthesis and price prediction, respectively. 3,043 images were obtained from the preprocessing of the image collection to separate out men's sneakers. The generator and discriminator that make up the GAN model were created to convert random noise into realistic sneaker designs. Based on the generated sneakers images, six regressors were used to forecast prices.

The result of the generated images showed close similarities with the actual input images. This was done using the Gini evaluation which produced a Z-test of 0.956. On the other hand, the prices of the models used to predict the prices did not perform as expected. The root mean squared error of the following regression models used for price prediction; Decision Tree, Lasso, Random Forest, Gradient Boost, Support Vector and K-Neighbours were 91.99, 82.28, 80.95, 87.06, 82.76 and 76.68 while the coefficient of determination on the regressors were -0.055, 0.156, 0.183, 0.055, 0.146 and 0.2666 respectively.

Findings show that GANs can speed up fashion design processes, and that the generated images can change over time in terms of realism and appeal. However, more improvement is needed to ensure price prediction accuracy. The significance of the study is that it provides a quick substitute for fashion design, making it available to people with limited design experience.

1. INTRODUCTION

Fashion as described by Fontanna and Miranda (2016) is a ‘complex social and cultural phenomenon with strong economic implications’ which means that fashion is influenced by a multitude of aspects of human existence, and it is highly interconnected into the social and cultural aspects of the society. Fashion, being an element of culture and social has been seen to be adopted and admired globally (Tortora, 2010). Due to this globalization, many top organizations like Addidas, Clarks, Zara and many more, have adopted fashion as their niche in business.

The process of fashion development involves two phases: the design phase and the production phase. The design phase has been seen to be expensive and time consuming. Due to the different variety of individuals’ taste in fashion, there is a need to improve and quicken the design phase of fashion development. One way in which this can be done is with the introduction of Generative Adversarial Networks (GANs) into the Fashion Industry.

Generative Adversarial Network (GAN) is a framework used in unsupervised machine learning to solve generative modeling problems (Goodfellow *et al.*, 2014). This is a framework which consists of 2 models: the generator and the discriminator. The work of the generator is to generate images and try to fool the discriminator into thinking the images created are real while the discriminator acts as a judge in determining if the image is real or fake.

There are different types of GAN. Some of them include; Conditional GAN(cGAN) (Mirza and Osindero, 2014), Deep Convolutional GAN(DCGAN) (Radford, Metz and Chintala, 2015), Style GAN (Karras and Aila, 2019), Disco GAN (Kim *et al.*, 2017) and many more.

My primary objective is to generate images of male sneaker designs using Deep Convolutional GAN and furthermore, predict prices of the generated sneakers. To achieve this, my aim is to generate fashionable and attractive male sneakers, provide a quick means to producing male sneaker designs and provide accurate price predictions for the generated sneaker designs based on prices on ecommerce platforms/ stores.

However, my research questions are as follows:

- Are the generated images fashionable, attractive and appealing?
- How fast does it take to generate male sneaker designs?
- How accurate are the price of the generated images, as compared to the sneakers on ecommerce platforms and fashion stores?

1.1. LITERATURE REVIEW

Fashion items generated with GANs can be seen in FashionGAN: Display your fashion design using Conditional Generative Adversarial Networks (Cui *et al.*, 2018) which was based on the conditional GAN architecture. Here, the sketch of the image as well as the specific fabric pattern image was put into the model after which the desired image is generated with the shape of sketch image and the texture of the fabric. The goal was to create a framework that can automatically generate images of clothing without the need for labor-intensive processes or knowledgeable user interactions during the design phase hereby increasing the efficiency of clothing design and build an architecture that can generate images with specific visual information. FashionGAN, alongside four other models namely; BicycleGAN (Zhu, Darrell, *et al.*, 2017), pix2pix (Isola *et al.*, 2017), MUNIT(Huang *et al.*, 2018), CycleGAN (Zhu, Park, *et al.*, 2017) and TextureGAN (Xian *et al.*, 2018) were trained on over 24,000 images. On human performance which was judged by 30 people, it was decided on the color accuracy and shape accuracy, the average overall was 4.381 on FashionGAN, 2.548 on pix2pix, 3.739 on BicycleGAN, 2.083 on CycleGAN, 2.441 on MUNIST and 3.138 on TextureGAN.

Furthermore, using Conditional GAN, Wu *et al.*, (2021) proposed ClothGAN, a generative adversarial network that generates a certain clothing known as Dunhuang which is popularly known in the Chinese culture. This GAN was combined along with a style transfer algorithm to get the desired result. A dataset of 52,908 Dunhuang images were used together with the FashionMNIST dataset which contains 60,000 images were used for this experiment. The Condition-GAN was used to generate clothing images based on the FashionMNIST which was used to train the GAN model, while the style transfer algorithm was trained with the Dunhuang dataset to add patterns to the generated images. The major problem faced here was the ability to generate fashion images which will have the Dunghan style.

Like the ClothGan, but with another approach of generating images, Salami, Oyewusi and Adekanmbi (2021) created AFRIGAN, an African Fashion Style Generator using Generative Adversarial Networks. The aim was to generate African clothing images using the Style-GAN, a type of GAN popularly known for generating high resolution images (Karras and Aila, 2019). A dataset containing 1600 images of African clothing was used to generate more clothing images using TensorFlow's implementation of the StyleGAN.

However, Deverall, Lee and Ayala (2017) change the focus from generating clothing designs to generating shoe designs. The goal in this experiment was to create a framework that can accept a condition, which in this case was the description or attribute of the shoes, and then the model would generate the shoes based on the condition given. This experiment was carried out using Conditional GAN after which classification was done to get the attributes. The classification done was broken down into functional classification and attribute classification. For functional classification, the target was to get at least an accuracy of 80%. A simple classifier and a GoogLeNet classifier were performed and a result of 48% and 97% achieved respectively. For, the binary attribute classification which was to identify if the shoe is pointy or not, VGG Net was used, and it resulted to achieving an accuracy of 58%.

2. DATASET

Two fashion datasets were used for this experiment. One for sneakers image generation and the other for price prediction of the generated sneakers images.

The fashion dataset used for image generation was gotten from Kaggle**,** a data science competition platform and an online community of data science and machine learning practitioners under Google LLC. Kaggle is also trusted by some of the largest companies in the world such as Mest (formerly known as Facebook) and Walmart (https://www.kaggle.com/). Therefore, datasets gotten from Kaggle are deemed reliable. This dataset was uploaded by a Data Engineer called, Param Aggarwal.

The fashion dataset contained an image folder of 44,441 fashion images ranging from footwears, accessories, apparels, personal care, sporting goods. Alongside the image folder were 2 documents namely; image.csv, which contains 2 columns (filename and the link) of the images and styles.csv which contains 10 columns (id, gender, masterCategory, subcategory, articleType, baseColour, season, year, usage and productDisplayName) of the images. The information from each feature was enough to describe the images. This made sorting of the images easier and suitable to carry out the task of generating more images. All the images in the dataset are of the same width and height; 1800px by 2400px.

The second dataset used for price prediction was gotten from Data.World, a data catalog platform/ open data community for AI engineers and data scientists/analysts. It is also a Certified B Corporation and public benefit corporation (https://data.world/company/about/). The fashion dataset (Data.World, 2017) which was uploaded by Datafiniti, contains a csv document of 48 columns, 19,388 rows of various fashion items.

2.1.DATA PREPROCESSING

The dataset obtained from Kaggle was a mixture of several fashion items. The categories of fashion items used to sort out the men sneakers were casual and sport, which resulted to a total of 3,166 images.

However, there was a need for more cleaning, so the following had to be done:

- The images had a white/colourless/clear background.
- The images were in the same direction.
- The images were not in pairs.
- The images did not contain any human body part.

The conditions were properly checked for using the human eyes and that further reduced the total number of images for men's sneakers to 3,043, which were used as the input images.

The second dataset from Data.World, was also a mixture of several fashion items. Unlike the dataset obtained from Kaggle, this dataset looked more like a dataset from an ecommerce platform. It contained more columns like the price, brand, date_added, date_updated, and more. To successfully carry out the price prediction of the generated images, the important columns are the price and the image URL, which will be used to download the images from the internet.

Images were downloaded and sorted to get the desired sneaker images to look similar to the images from the first dataset. 1,602 sneaker images were retrieved.

Figure 1. This is a sample of how all the input images look before they are passed into the generator model. The images are with clear backgrounds, not in pairs, do not possess any human body part and are all facing the same direction.

3. METHODOLOGY

3.1.IMAGE GENERATION

After preprocessing of the dataset, the input images were ready for the discriminator. They were resized to a desired dimension of 128 x 128 after which they were converted to arrays of numbers. The numbers were within the range of 0 to 255. Normalization was done to reduce the range of the images between 0 and 1 after which conversion to tensors occurred. The converted images where further batched with a batch size of 128.

The deep convolutional GAN (DCGAN), which is a type of GAN model, is used here to generate images. In this GAN model, the Conv2DTranspose is used to scale up the input images. Unlike the upsampling2D which also scales up images, the Conv2DTranspose also possesses parameters that learn from the dataset using kernels as it scales up. Hence the name, 'deep convolutional'.

In this model, the generator which receives an input of random noise with a dimension of 300, contains 5-layer blocks, the first layer block which is the input layer, is a fully connected layer that takes in an input and reshapes into a size of 32 x 32 which then passes through the hidden layer that consists of two more conv2DTranspose layers. Both layers help to transform the input to the a shape of 128 x 128. However, the final layer which is a Conv2DTranspose with a tanh activation, serves as the output layer that converts the input to an image with 3 channels. In all layers except the input layer and the output, there is an introduction of two layers; a layer with batch normalization and an activation layer with LeakyReLU. All Conv2DTranspose layer contained the stride attribute as well as a 5 x 5 kernel.

The Discriminator on the other hand, contains an input layer that receives the desired input shape of 128 x 128 x 3, a hidden layer which contains a Conv2DTranspose layer and output layer which contains a flatten layer and a fully connected layer. Here, all layers contain the LeakyReLU activation layer and a dropout regularization layer of 0.3.

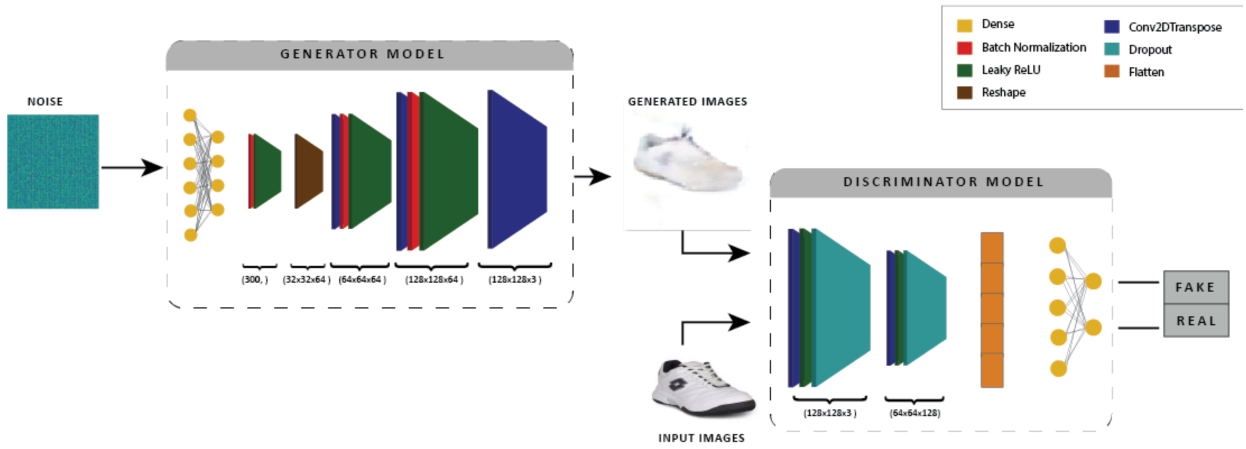

Figure 2. This illustrates how the noise input is passed through the generator model and comes out as generated images which is then passed into the discriminator model where it is classified or criticized as real or fake images.

3.2.PRICE PREDICTION

To be able to predict the price of the generated sneaker images, it was decided that the 1,602 images will be trained alongside the prices assigned to it. To do this, the images were also resized to a dimension of 128 x 128, after which they were converted to numpy arrays and normalized like the dataset obtained from Kaggle.

The images were further reshaped to a 1D numpy array and trained alongside the prices. The feature variable was the set of converted images as 1D numpy arrays and the target variable was the price. The dataset was then divided into training and testing. The following models used to train the dataset include Lasso, Kn-Neighbour, Gradient Boost, Support Vector Regression (Drucker *et al*., 1996), Decision Tree Regression and Random Forest Regression (Breiman, 2001).

4. RESULT

4.1.GAN MODEL

The model designed was made to generate 16 sneaker images and it was made to train over 3000 epochs. The generated images at the first epoch were more of noise with a generator loss of 0.690 And a discriminator loss of 0.697.

At 232 epochs, the generated images began to look like drawn sketches. Although the clarity of the images was not perfect, it was obvious the images generated were sneaker images. The more the images were trained over epochs, the better the images.

At 1409 epochs, the generated images began to look more realistic. Colours and image depth (which was as a result of the input images being 3 channels) were noticed. At this point, the images generated had a discriminator loss of 1.321 and a generator loss of 0.330. It was also generated in 98 seconds.

Figure 3. This illustrates the evolution of the images across several epochs. Image A shows the generated images at epoch 1 where it is seen as a noise. It evolves through to Image B at epoch 232 where the images begin to look like sketches. Further evolution to epoch 1795 before it gets to the final image at epoch 3000.

4.1.1. LOSSES

In this GAN model, we are concerned with the discriminator loss and the generator loss. It should be noted that when a discriminator loss reduces, it simply means that there is an improvement in the discriminator's ability to identify a real image and a fake image. Also, if the generator loss reduces, it means there is an improvement in the generator's ability to generate images that are identified as real images by the discriminator.

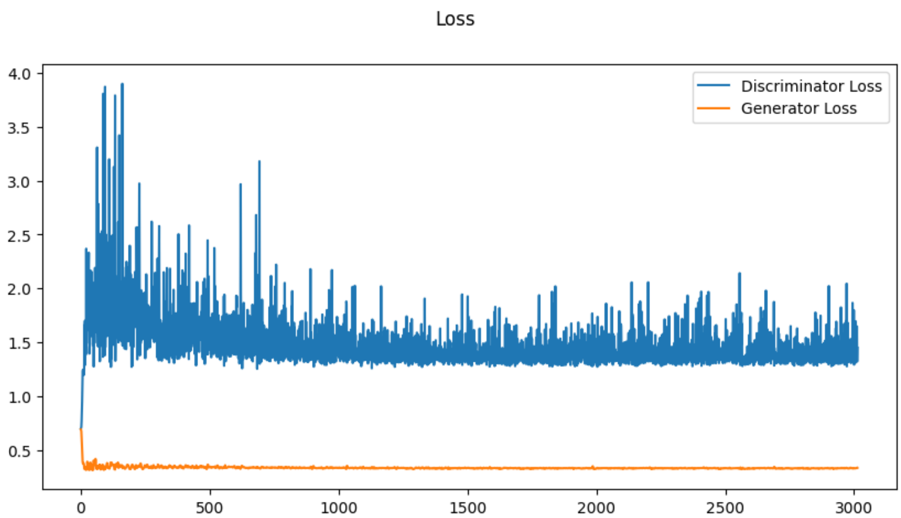

Figure 4. The plot illustrates the convergence between the discriminator loss and the generator loss across 3000 epochs. This shows there is a balance between generator and the discriminator. Therefore, the GAN model has learned enough.

	Epoch	Time (in seconds)	Discriminator Loss	Generator Loss
0	Epoch 1	97	0.697	0.690
1	Epoch 2	100	0.702	0.685
2	Epoch 3	103	0.716	0.673
3	Epoch 4	100	0.767	0.630
4	Epoch 5	101	0.849	0.567
…	…	…	…	…
2996	Epoch 2996	95	1.439	0.333
2997	Epoch 2997	96	1.330	0.332
2998	Epoch 2998	96	1.648	0.331
2999	Epoch 2999	96	1.321	0.332
3000	Epoch 3000	98	1.449	0.334

Table 1. Top 5 and bottom 5 losses of both discriminator and generator.

4.1.2. EVALUATION

The evaluation metrics chosen for this GAN model are the Gini Evaluation (Gini, 1936) and SSIM Evaluation (Wang et al, 2004). Gini evaluation also known as Gini coefficient in images is used to measure the distribution of pixel values or colour values in images while the SSIM evaluation is the measure of similarities between two images.

4.1.2.1. GINI EVALUATION

The Gini coefficient ranges between 0 and 1 with 0 meaning it is perfect equality of the pixel values while 1 meaning it is perfect inequality of the pixel values. The similarity between the generated images and the input images was concluded with a z-test result of 0.956.

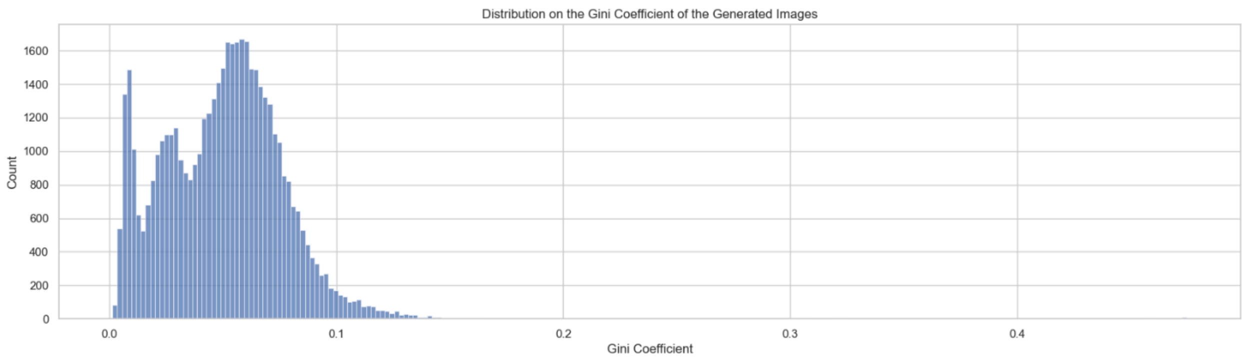

Figure 5. shows the distribution of the Gini Coefficient of all the generated images. This illustrates that about 90% of the generated images are within the coefficient of 0.0 and 0.1. However, the images are close to having a perfect equality.

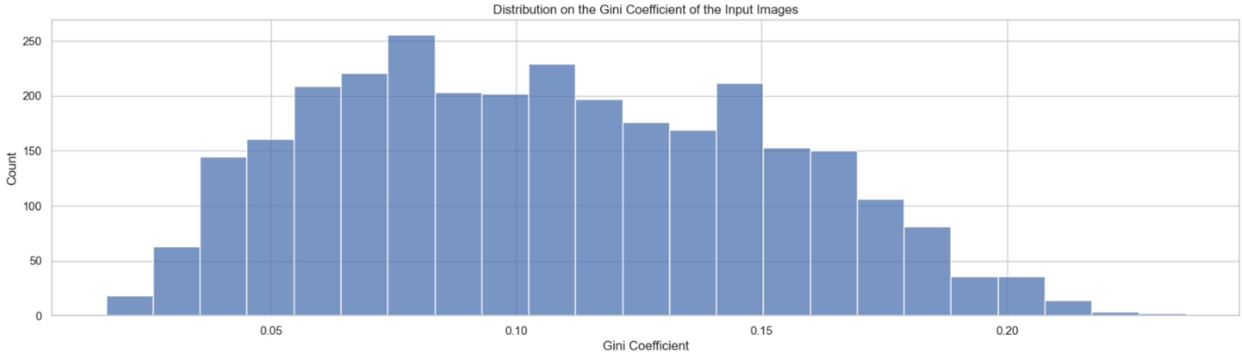

Figure 6. shows the distribution of the Gini Coefficient of all the input images. This illustrates that about 90% of the generated images are within the coefficient of 0.0 and 0.2.

4.1.2.2.SSIM EVALUATION

Like Gini, the SSIM evaluation ranges between 0 and 1 with 0 signifying there is no similarity and 1 signifying it is a perfect similarity between two images. However, the average SSIM between the input images and the generated images is 0.659 while the standard deviation is 0.037.

4.2.PRICE PREDICTION

To predict prices, the ecommerce dataset with 1,602 sneakers images was trained on 6 regressor models: Decision Trees, Random Forest, Lasso, Gradient Boost, Support Vector and K-Neighbour regressors.

	Test price	**Random Forest Prediction**	**K-Neighbour Prediction**	**Gradient Boost Prediction**	**Lasso Prediction**	**Support Vector Prediction**	**Decision Tree Prediction**
0	130.00	66.92	67.22	65.63	82.11	53.62	105.00
1	33.99	73.95	80.15	74.20	70.08	40.82	75.00
2	115.00	81.20	118.02	81.10	11.20	50.45	54.99
3	27.99	64.22	77.42	56.73	101.34	103.15	70.00
4	87.35	78.39	75.58	90.84	15.54	82.61	71.84

Table 2. Comparison between price values of the test dataset and price values predicted by 6 regressor model.

Generated Images	Predicted Price (Random Forest)	Predicted Price (K-Neighbour)	Predicted Price (Gradient Boost)	Predicted Price (Lasso)	Predicted Price (Support Vector Regressor)	Predicted Price (Decision Tree Regressor)
img_3000_1.jpg	94.35	80.38	68.03	71.88	75.11	79.95
img_3000_2.jpg	79.77	68.10	100.81	70.76	74.91	34.99
img_3000_3.jpg	141.53	78.78	186.30	121.83	99.01	225.00
img_3000_4.jpg	122.66	76.79	106.08	81.87	58.87	109.99
img_3000_5.jpg	93.65	79.24	98.52	97.68	112.26	142.95
img_3000_6.jpg	73.19	79.03	62.72	80.70	72.28	69.05
img_3000_7.jpg	91.74	80.30	118.24	75.90	71.12	79.95
img_3000_8.jpg	98.59	68.92	78.12	11.46	17.64	520.00
img_3000_9.jpg	85.64	79.77	65.36	67.13	56.35	69.05
img_3000_10.jpg	86.61	64.52	87.92	24.34	42.29	130.00
img_3000_11.jpg	65.99	72.55	65.20	85.47	53.28	39.99
img_3000_12.jpg	89.60	80.02	104.86	91.66	83.09	69.05
img_3000_13.jpg	92.39	64.58	71.23	46.35	81.96	57.14
img_3000_14.jpg	80.00	74.54	97.91	92.33	85.76	39.99
img_3000_15.jpg	63.27	79.98	61.69	75.51	79.63	39.95
img_3000_16.jpg	73.45	76.41	62.39	65.66	74.06	57.14

Table 3. Price prediction of all 16 generated images using 6 regressor models.

4.2.1. EVALUATION

The evaluation metrices used for price prediction are the R2 Score, Mean Absolute Error and Root Mean Squared Error (Tatachar, 2021). The evaluation are as follows:

	Decision Tree	Lasso Regression	Random Forest Regression	Gradient Boost Regression	Support Vector Regression	K-Neighbours Regression
R2 Score	-0.055	0.156	0.183	0.055	0.146	0.2666

Mean Absolute Error (MAE)	51.965	50.908	41.033	42.884	48.333	41.587
Root Mean Squared Error (RMSE)	91.999	82.282	80.951	87.059	82.763	76.687

Table 4. R2 score, Mean Absolute Error, and Root Mean Squared Error evaluation metrics across 6 regressor models.

5. DISCUSSION

a) How realistic and appealing are your images?

To determine how realistic and appealing the images are, I propose the use of an SSIM evaluation and a survey. Unlike the SSIM evaluation highlighted in the result which compared the input image and the output image, the SSIM evaluation used here should compare the generated image with a real image with an evaluation between 0 and 1 where 0 represents no similarity and 1 represents perfect similarity.

On how appealing the images are, this will be dependent on the preference of humans through a survey. In my opinion, the images can be better and look more appealing.

b) What was the performance of GAN model?

The performance of my Deep Convolutional GAN model is dependent on the time and losses which are the discriminator loss and generator loss. From my result above, the time taken to generate an image varied between 83 to 107 seconds. However, there was a reduction with the losses of the generator with the loss starting at 0.670 at the first epoch and ending at 0.334 at the final epoch. This indicates that there was improvement with the generation of images from the generator. At the discriminator, there was a struggle with the criticism of the generated images as the discriminator loss fluctuated. This meant that the discriminator was having a tough time criticizing the generated images.

c) What are the steps taken to optimize the performance of GAN model to get better and appealing images?

To get better images, the following steps were taken:

- Ensured that the learning parameters were enough. This was done with the help of strided convolutions,
- Increased the latent space dimension to 300.
- Addition of the dropout regularization to prevent overfitting.

- Ensured that the binary cross entropy was the loss function used at the discriminator because the discriminator will perform a binary classification of sorting the generated images into real and fake images.

d) How accurate are your price predictions?

The prediction of prices was not as accurate as expected. Despite using over 6 regression models in the prediction of prices, the mean squared error of all regressor models were rather too large. However, to resolve this, I would determine if there could be outliers in the prices. This can be done with the Multiple of IQR Test or the Grubb's Test.

Another thing that can be done is Feature Engineering. The only feature variable trained against the price was the converted images in 1D numpy arrays. More features can be included to train alongside the images.

e) Why should your GAN model be considered and how can it impact the society?

The impact of the GAN model can be outlined in the following:

- **Time**: The GAN model was created to be able to generate 16 sneakers image designs. It took approximately four days to generate these images. However, this is dependent on the graphics performance unit of the computer system used to train the model.

- **Little or No Knowledge:** According to Indeed, (Indeed.com, 2023), a top job platform, one would need to study designs and probably enroll into a shoe design program. With my GAN model, you require no knowledge of shoe designing.

f) Am I surprised at the outcome?

The GAN model was created to accept several images, learn from the images through its parameters and generate more images based on what has been learnt. Therefore, the outcome of the generated images was expected and not surprising. However, I was surprised that the regression models were not able to accurately predict the prices on the test dataset as expected.

CONCLUSION

In summary, the application of Generative Adversarial Networks (GANs) in the fashion industry, specifically for male sneaker design generation, has shown promise. The Deep Convolutional GAN (DCGAN) model exhibited improved realism and appeal over training epochs, as indicated by discriminator and generator losses. However, comprehensive evaluation using metrics like SSIM, Gini and human preference surveys is needed.

Optimization steps, including increasing latent space dimensions, implementing dropout regularization, and selecting appropriate loss functions, were employed to enhance the model's performance. Despite advancements, further assessment and refinement are required.

Regarding price prediction based on generated images, regression models displayed potential for improvement. Incorporating outlier detection and feature engineering could enhance the accuracy of sneaker price predictions.

The societal impact of this GAN model lies in its potential to streamline the fashion design process, particularly for sneakers. It offers an efficient means of generating diverse and appealing designs without extensive design expertise, potentially democratizing fashion design and enabling broader participation in the creative process.

FUTURE WORK

Due to the result obtained from the price prediction, I would like to do further feature extraction by increasing the number of features to improve the accuracy of prices prediction. In addition to this, I would like to try out other types of GAN in the generation of sneakers.

REFERENCES

Breiman, L. (2001) *Random forests*. Machine learning, 45, 5-32. https://doi.org/10.1023/A:1010933404324

Cui, Y.R., Liu, Q., Gao, C.Y. & Su, Z. (2018) FashionGAN: display your fashion design using conditional generative adversarial nets. *In Computer Graphics Forum*, 37(7), 109-119.

Data.World. (2017) *https://data.world/datafiniti/mens-shoeprices/workspace/file?filename=7004_1.csv*

Deverall, J., Lee, J. & Ayala, M. (2017*)* Using generative adversarial networks to design shoes: the preliminary steps. *CS231n in Stanford*.

Drucker, H., Burges, C.J., Kaufman, L., Smola, A. & Vapnik, V. (1996) Support vector regression machines. *Advances in neural information processing systems*, 9.

Fontana, G.L. & Miranda, J.A. (2016) The business of fashion in the nineteenth and twentieth centuries. *Investigaciones de Historia Económica-Economic History Research*, 12(2), 68-75.

Gini, C. (1936) On the measure of concentration with special reference to income and statistics. *Colorado College Publication, General Series*, 208(1), 73-79.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014) Generative adversarial nets. *Advances in neural information processing systems*, 27.

Huang, X., Liu, M.Y., Belongie, S. & Kautz, J. (2018) Multimodal unsupervised image-to-image translation. *In Proceedings of the European conference on computer vision (ECCV)*, 172-189.

Indeed. (2023) How to become a shoe designer in 8 steps. *https://www.indeed.com/career-advice/finding-a-job/how-to-become-shoe-designer*

Isola, P., Zhu, J.Y., Zhou, T. & Efros, A.A. (2017) Image-to-image translation with conditional adversarial networks. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 1125-1134.

Kaggle. (2018) https://www.kaggle.com/datasets/paramaggarwal/fashion-product-images-dataset

Karras, T., Laine, S. & Aila, T. (2019) A style-based generator architecture for generative adversarial networks. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4401-4410.

Kim, T., Cha, M., Kim, H., Lee, J.K. & Kim, J. (2017) Learning to discover cross-domain relations with generative adversarial networks. *In International conference on machine learning*, 1857-1865.

Mirza, M. & Osindero, S. (2014) Conditional generative adversarial nets. *arXiv preprint, arXiv:1411.1784.*

Radford, A., Metz, L. & Chintala, S. (2015) Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint, arXiv:1511.06434*.

Salami, M., Oyewusi, W. & Adekanmbi, O. (2021) AFRIGAN: African fashion style generator using Generative Adversarial Networks (GANs). *In Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS)*, 1-3.

Tatachar, A.V. (2021*)* Comparative Assessment of Regression Models Based On Model Evaluation Metrics. *International Journal of Innovative Technology and Exploring Engineering*, 8(9), 853-860.

Tortora, P.G. (2010) History and Development of Fashion. *Berg Encyclopedia of World Dress and Fashion*, 10.

Wang, Z., Bovik, A.C., Sheikh, H.R. & Simoncelli, E.P. (2004) Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4), 600-612.

Wu, Q., Zhu, B., Yong, B., Wei, Y., Jiang, X., Zhou, R. & Zhou, Q. (2021) ClothGAN: generation of fashionable Dunhuang clothes using generative adversarial networks. *Connection Science*, 33(2), 341-358.

Xian, W., Sangkloy, P., Agrawal, V., Raj, A., Lu, J., Fang, C., Yu, F. & Hays, J. (2018) Texturegan: Controlling deep image synthesis with texture patches. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 8456-8465.

Zhu, J.Y., Zhang, R., Pathak, D., Darrell, T., Efros, A.A., Wang, O. & Shechtman, E. (2017) Toward multimodal image-to-image translation. *Advances in neural information processing systems*, 30.

Zhu, J.Y., Park, T., Isola, P. & Efros, A.A. (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. *In Proceedings of the IEEE international conference on computer vision*, 2223-2232.