iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** IV **Month of publication:** April 2023

DOI: https://doi.org/10.22214/ijraset.2023.50469

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue IV Apr 2023- Available at www.ijraset.com

Fashion Clothes Generation Systemusing Deep Convolutional GAN

Ms. M. Deepthi[1], V. Meghana[2], N. Harshini[3], V. Sahana[4], B. Tejasri[5]
Department of Information Technology,GVPCEW,

Abstract: ***The fashion industry has evolved into one of the most powerful industries in the world as a result of modernization. Before the middle of the 19th century, almostall types of clothing were made specifically for each person, either at home or on demand from dressmakers or tailors. Technological advancements such as the development ofartificial fibers, and nylon, as well as new dyeing and fabric cuffing processes, have given designers more creative flexibility. Likewise, the fashion industry has emerged various buying options like e- commerce platforms these days rather than the traditional approach. Where, some websites use automatic pattern generation in place of the conventional method (clothing designs). However, these websites are not likely to make high-end apparels accurately, which is why we propose to generate new fashionable clothes and develop a web application that generates high-end fashion apparels based on the training dataset by taking input from the users (the number of images that need to be generated by the model) using GAN technology, and letting the user choose colors for the generated apparels. GAN, short for Generative AdversarialNetworks, is a type of deep learning model that is used for generating synthetic data that is similar to the original data. It is composed of two neural networks - the generator and the discriminator - that are trained simultaneously to create and evaluate the synthetic data. For the purpose of creating high-quality fashion images, we suggest using Deep Convolutional Generative Adversarial Networks (DC-GANs). A deep learning technique called DC- GANs using convolutional layers in an adversarial network to produce images of a particular type. The "color palette" feature is implemented using a basic Image Processing Technique such as object color translation. Once the object is segmented, its color can be modified using various color transformation techniques such as RGB to HSV conversion or color balance adjustment.***

Keywords: Neural Network, Generative Adversarial Networks (GANs), DC GAN, generator, discriminator, convolutional, deconvolutional, batch normalization, loss function, activation function, color translation techniques.

I. INTRODUCTION

The rise of new technologies has produced new means of livelihood like online shopping, multimedia, entertainment, gaming, and advertising. One of the sectors greatly impactedby the new paradigm is the fashion industry. Every season, new trends enter the fashion world, which is continuously changing. To stay relevant, fashion designers and retailers must be able to generate new and trendy Clothes that appeal to customers. Generating fashionable clothes that cater to different tastes and preferences using cutting-edge technologysuch as virtual reality, 3D printing, and machine learning canhelp retailers and designers stand out in the market. The online applications for fashion alone have been immensely developed. However, the applications that will assist designers in reducing the work of creating new pattern-basedclothes based on users' interests are nonexistent. Even thoughsome applications are there, they won't generate high- resolution patterns.

The aforesaid problems can be resolved by employing GAN (Generative Adversarial Networks) technology. When there isa web application that incorporates the Generation of Fashionable Clothes (with high-resolution photographs) and making it as an E-Boutique (similar to an e-commerce website) that displays various colors for each piece of apparelto satisfy users and allow them to see more shades of apparel. Using image processing techniques, such as the fundamental image translation technique, the color changing of apparels can be accomplished.

A Generative Adversarial Network (GAN) is a class of machine learning systems invented by Ian Good fellow and his colleagues in 2014.GAN's consist of two main components: a generator and a discriminator. The generator creates new, synthetic data that is similar to the real data it istrained on. It does this by taking a random input (called a latentvector) and producing an output that should be similar to thereal data. The generator is typically implemented using a neural network. The discriminator is a model that is trained todistinguish between the synthetic data produced by the generator and real data. It evaluates the output of the generator and assigns a probability that it is real. The discriminator is also implemented using a neural network.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue IV Apr 2023- Available at www.ijraset.com

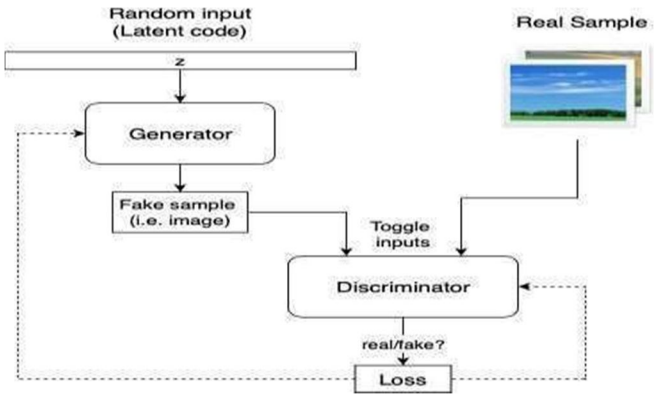


Figure1: A Diagram of how a GAN works

II. LITERATURE SURVEY

The obsession with outfits has evolved over the years, and there have emerged various buying options like e-commerceplatforms these days rather than the traditional approach. However, the availability of the new trends or choices requires further user interaction in order to generate fashionable clothes. We propose to generate new fashionable clothes using Deep Convolutional Generative Adversarial Networks (DC-GAN) and develop an application where the user needs to give input to the interface, and the system will generate those counts of images and display them. Following generation, the user canalso experience the color palette of the selected outfit. The DCGAN is a GAN architecture where the discriminator and generator are defined with Convolutional Neural Networks (CNNs). CNNs, or Convolutional Neural Networks, are a type of neural network commonly used for image and video processing. They are designed to recognize spatial patterns ininput data, such as images, by using a hierarchical arrangement of filters, also called convolutional layers, that learn to extract and abstract features from the data. Using an image colour translation technology, various colours for a particular article of clothing are represented. Image color translation is a popular technique used in various applications such as image segmentation, image enhancement, and color correction. It can be used to modify the color of an image in a variety of ways. One common method is to adjust the hue channel of the image. This method involves shifting the hue values of the image to produce a different color palette.

A. Neural Networks

The structure and operation of the human brain serve as the basis for the machine learning model known as "NeuralNetworks." Layers of interconnected "neurons" that process and send information make up their structure. Neural networks are trained using large amounts of data, and they areable to learn and make predictions or decisions without beingexplicitly programmed to perform a specific task. When we are dealing with images, we use convolutional neural networks (CNN). A convolutional neural network (CNN) is a type of neural network that is particularly well suited for image recognition and processing tasks. CNNs aremade up of multiple layers, including an input layer, multiple hidden layers, and an output layer. The key building block of a CNN is the convolutional layer. In a convolutional layer, a set of filters (also called kernels or weights) is used to detect different features in the inputimage. These filters are passed over the image in a sliding window fashion, and a dot product is taken between the filter and the region of the image that it is currently "lookingat." This produces a feature map that highlights the presenceof certain features in the input image. After one or more convolutional layers, a pooling layer is often used to reduce the spatial size of the feature maps. This is done by applyinga pooling operation (such as max pooling) to smallsubregions of the feature map. This has the effect of reducing the number of parameters in the model as well as making it more robust to small translations of the input image. Finally, a fully connected layer is used to interpretthe features extracted by the previous layers and produce an output. The output is typically a probability distribution overa set of classes (e.g., "dog," "cat," "car").

B. DC-GAN Explanation

DCGAN stands for Deep Convolutional Generative Adversarial Network. It is a type of Generative Adversarial Network (GAN) that uses convolutional neural networks (CNNs) to generate high-quality images. The key idea behind DCGAN is to use CNNs to learn a mapping from a latent space (typically a high-dimensional random noise vector) to the space of realistic images.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

The generator network in DCGAN consists of a series of deconvolutional layers that take the input noise vector and gradually produce an output image. The discriminator network, on the other hand, consistsof a series of convolutional layers that process the input image and produce a probability score indicating whether the image is real or fake.During training, the generator and discriminator networks are trained in a minimax game wherethe generator tries to produce images that are indistinguishable from real images, while the discriminator tries to correctly classify between real and generated images. This feedback loop helps the generator to learn to produce more realistic images, and the discriminator to learn to better distinguish between real and generated images. One of the main advantages of DCGAN is that it can generatehigh-quality images with realistic details, such as sharp edges, intricate textures, and fine details. This is achieved byusing a combination of convolutional and deconvolutional layers, which can effectively capture and represent the spatialstructure of images. DCGAN has been used for a variety of image generation tasks, including generating realistic imagesof faces, landscapes, and objects. It has also been used for image style transfer, image inpainting, and other image processing tasks. Overall, DCGAN is a powerful tool for generating high-quality images, and it has become an important component of many state-of-the-art deep learning models.

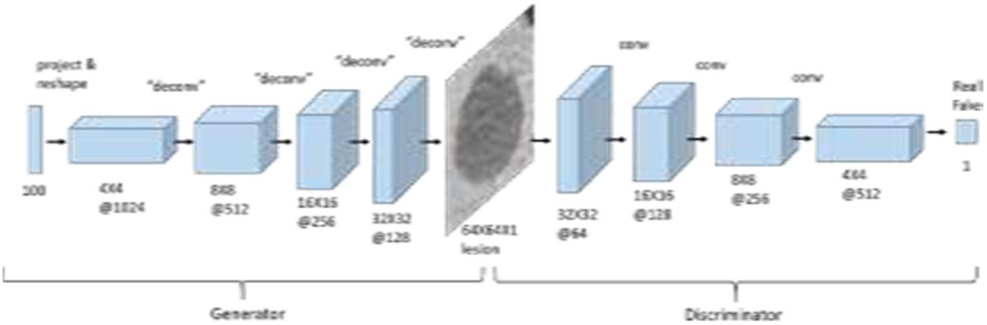


Figure2: DC-GAN Architecture

C. *Batch Normalization*

Batch normalization is used in GANs to stabilize the training process and improve the performance of the generator and discriminator networks. The basic idea behind batch normalization is to normalize the activations of a layer by adjusting and scaling the activations. It is a technique used indeep learning to normalize the inputs to a layer. It helps to improve the stability and performance of neural networks byreducing the internal covariate shift, which is the change in the distribution of the inputs to a layer caused by the updatesin the parameters of the previous layers. This is done by computing the mean and standard deviation of the activationsfor a mini-batch of data, and then using these statistics to shiftand scale the activations. The shift and scale parameters are learned during training and are part of the model. Batch normalization can be applied to any type of layer, but it is particularly useful for deep networks and for layers with a large number of inputs. Additionally, batch normalization makes it possible to use higher learning rates and reduce the dependence on the initialization of the weights, which furtherimproves the training stability and performance of GANs.

D. *Activation Functions*

There are four commonly used activation functions in DCGAN generator and discriminator: sigmoid, tanh, ReLU and leakyReLU as illustrated in Figure:

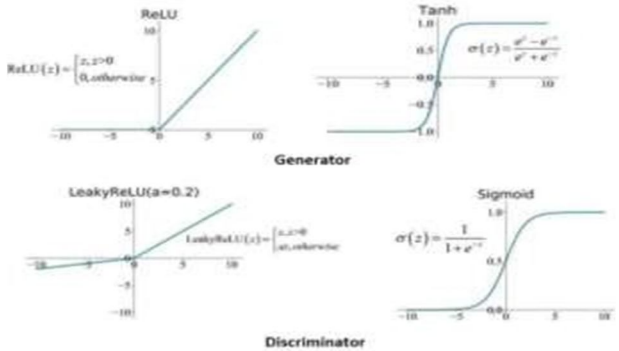


Figure3: Activation Functions in DCGAN

1) *Sigmoid:* Squashes the number to 0(fake) and 1(real). Since the DCGAN discriminator does binary classification, we use sigmoid in the last layer of D.
2) *Tanh (Hyperbolic Tangent):* It is also s-shaped like sigmoid;in fact, it's a scaled sigmoid but centered at 0 and squashes the input value to [-1, 1], we use tanh in the last layer of G. This is why we need to preprocess our training images to therange of [-1, 1].
3) *ReLU(Rectified Linear Activation):* Returns 0 when the input value is negative; otherwise, it returns the input value. ReLU activation is used for all layers in G except for the output layer, which uses tanh.
4) *LeakyReLU:* Similar to ReLU except when the input value is negative, it uses a constant alpha to give it a very small slope. As suggested, we set the slope (alpha) as 0.2. We use LeakyReLU activation in D for all layers except for the last layer.

E. Loss Function

In the context of GANs, the BCE loss is used to calculate the dissimilarity between the predicted probability distributions of the real and fake images. The goal of the discriminator in a GAN is to correctly classify whether a given input is real or fake. In order to do this, it uses a binary cross-entropy loss function to compare its prediction (either real or fake) to the true label (1 for real images and 0 for fakeimages). The discriminator's goal is to minimizethis loss, which it does by adjusting its parameters to better distinguish between real and fake images. On the other hand, the generator's goal is to generate images that are similar to the real images. Its loss function is the opposite of the Discriminator, it is trying to maximize the BCE loss functionvalue by generating images that are similar to thereal images so that the discriminator will classify it as a real image. The formula for binary cross-entropy loss function used in DCGAN is as follows:

L_D = -[y * log(D(x)) + (1 - y) * log(1 - D(G(z)))]

where:

L_D is the binary cross-entropy loss function used in thediscriminator network.

y is the true label (either 0 or 1).

D(x) is the discriminator's output when the input is a real image x.

G(z) is the generator's output when the input is a random noise z.

D(G(z)) is the discriminator's output when the input is a fakeimage generated by the generator log is the natural logarithmfunction.

Intuitively, the binary cross-entropy loss function measures how well the discriminator is able to correctly classify real and fake images. When the discriminator is given a real image, the true label is 1, and the loss is proportional to howfar the discriminator's output is from 1. When the discriminator is given a fake image generated by the generator, the true label is 0, and the loss is proportional to how far the discriminator's output is from 0. The goal of training the discriminator network in DC GAN is to minimizethis binary cross-entropy loss function.

F. Image Colour Translation Technique

In order to represent different colors for a selected apparel,image color translation technique is used. Image colortranslation in image processing refers to the process of converting an image from one color space to another. A color space is a way of representing colors using a mathematical model. Common color spaces include RGB (red, green, blue), CMYK (cyan, magenta, yellow, black), HSV (hue, saturation, value), and YUV (luma, chroma). The processstarts by loading an image in BGR (Blue, Green, Red) colorspace, andthen converting it to the HSV color space. The huechannel is extracted from the HSV image, and then the huevalue is changed by a certain amount (in the example code, itis increased by 10 for every iteration) and wrapped around tofit within the range of 0-180 (since hue values range from 0-360 degrees in the HSV color space, but OpenCV representsit ina range of 0-180). After modifying the hue channel, it ismerged back with the saturation and value channels to createa new HSV image. This new image is then converted back tothe BGR color space, resulting in an image with the samebrightness and saturation as the original, but with a differenthue.This process is repeated for multiple iterations withdifferent hue values, resulting in a sequence of images withdifferent colors.

III. DATASET

The dataset considered is "Deep Fashion"

Table 1

Dataset Size	8,00,000 images
Dimension	256x256
Image Size	Kb
Image Type	JPG

1) DeepFashion dataset contains around 800K diverse fashion images with their rich annotations (46 categories, 1,000 descriptive attributes, bounding boxes and landmark information) ranging from well-posed product images to real-world-like consumer photos.
2) Dataset link: https://www.kaggle.com/datasets/pawakapan/deepfashi on

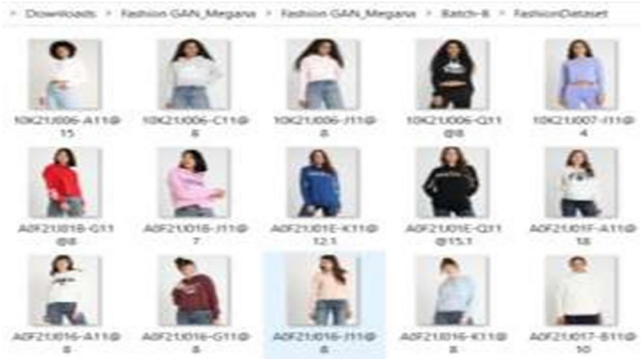


Figure4: Sample dataset downloaded

IV. ARCHITECTURE

There are a few guidelines that are commonly followedwhendesigning a DCGAN architecture:

1) Replace pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator): Pooling layers are often used in traditional CNN architectures to reduce the spatial size of the input. However, in GANs, pooling layers can cause artifacts in the generated images. Strided convolutions in the discriminator and fractional-strided convolutions in the generator are used instead to downsample orupsample the image while maintainingspatial resolution.
2) Use batch normalization in both the generator and thediscriminator: Batch normalization is a technique thathelps to stabilize the training process by normalizingthe inputs to each layer. In DCGAN, batch normalization is applied to both the generator and thediscriminator, which helps to speed up convergence andreduce the risk of overfitting.
3) Remove fully connected hidden layers for deeper architectures: Fully connected layers are often used in traditional neural network architectures, but they can lead to overfitting in GANs. In DCGAN, fully connected layers are removed in favor of convolutional layers, which allows for deeper architectures withoutthe risk of overfitting.
4) Use ReLU activation in the generator for all layers except for the output, which uses tanh: ReLU activation functions are commonly used in CNNs because they are computationally efficient and help to prevent the vanishing gradient problem. In DCGAN, ReLU is used for all layers in the generator except forthe output layer, which uses tanh to ensure that the pixel values of the generated image are in the range of-1 to 1.
5) Use LeakyReLU activation in the discriminator for alllayers: LeakyReLU activation functions are a modification of the standard ReLU function that allowfor a small, non-zero gradient when the input is negative. This helps to prevent the discriminator from getting stuckin a local minimum and improves the stability of the training process.

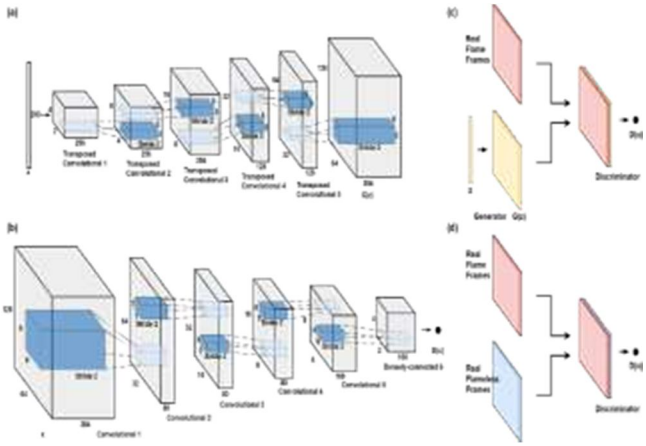


Figure5: DC-GAN Architecture

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue IV Apr 2023- Available at www.ijraset.com

A. Generator Architecture

The generator neural network in a DC GAN (Deep Convolutional Generative Adversarial Network) is responsible for producing new, realistic images by transforming a noise vector into a high-dimensional image. Here is a more detailed explanation of the internal processof the generator neural network in a DC GAN:

1) *Input Layer:* The input noise vector, usually a random vector of a fixed length, is fed into the input layer of the generator network.
2) *Hidden Layers:* The input noise vector is then passed through a series of hidden layers, each of which consists ofa convolutional layer, followed by batch normalization andan activation function. The convolutional layers apply a set of filters to the input noise vector, which detects and extracts features at different scales. Batch normalization helps to stabilize the distribution of the activations of the previous layer, which can speed up training and reduce over fitting. The activation functions, typically ReLU or LeakyReLU, introduce non-linearity into the network and help the network to learn more complex representations.
3) *Upsampling Layers:* After several hidden layers, the output feature map has a lower resolution than the target image. To upsample the feature map and increase its resolution, a series of deconvolutional layers are used. Theselayers transform the feature map into a higher- resolution feature map.
4) *Output Layer:* The final layer of the generator networktypically uses a Tanh activation function, which scales the output to be in the range of [-1, 1] or [0, 1] depending on theimage data. The output of this layer is the generated image that resembles the training data.
5) *Loss Function:* The generator is trained using an adversarial loss function, which measures how well the generator is able to fool the discriminator into classifying thegenerated images as real. This loss function helps to ensure that the generated images are realistic and resemble the training data.
6) *Backpropagation:* The error between the generatedimage and the real image is backpropagated through the network to update the weights and biases of the generator.

Overall, the generator neural network in a DC GANtransforms a low-dimensional noise vector into a high- dimensional image by learning a hierarchy of feature representations that capture the characteristics of the training data. The use of convolutional and deconvolutional layers, batch normalization, and activation functions help the generator to learn more complex representations and producehigh-quality, realistic images.

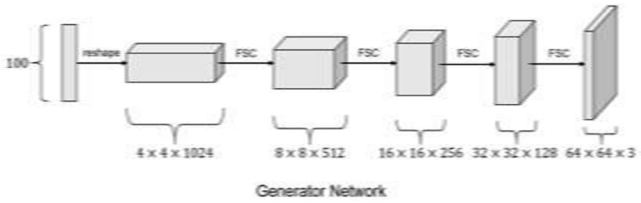


Figure6: Generator Architecture

B. Discriminator Architecture

The discriminator neural network in DC GAN is responsiblefor distinguishing between real images and fake images generated by the generator network. It takes an image as input, and produces a scalar output that represents the probability of the input being a real image.

The internal process of the discriminator neural network in DC GAN typically consists of a series of convolutional layersfollowed by a few fully connected layers. The convolutional layers are responsible for extracting features from the input image, while the fully connected layers are responsible for making the final prediction. Here is a more detailed explanation of the internal process of the discriminator neuralnetwork in DC GAN:

1) *Input Layer:* The discriminator neural network takes as input an image of size (height x width x channels), where channels typically correspond to the number of color channels in the image (e.g., 3 for RGB images). The input image is typically normalized to have pixel values in the range of [-1, 1] or [0, 1].
2) *Convolutional Layers:* The input image is passed through a series of convolutional layers, each of which applies a set of learnable filters to the input image to producea set of feature maps. The filters typically havesmall spatial extent (e.g., 3x3 or 5x5), and the numberof filters in each layer typically increases as we go deeper into the network. The filters are applied using a convolution operation, which involves sliding the filters over the input image and computing the dot product between the filter and the local region of the input.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

3) *Activation Function:* After each convolutional layer, anactivation function is applied to the feature maps. In DC GAN, the leaky ReLU activation function is typically used for all layers of the discriminator network except forthe last layer, which uses the sigmoid activation function.
4) *Strided Convolution:* Instead of using pooling layers, strided convolutions are used in the discriminator network todown sample the feature maps. Strided convolutions are similar to regular convolutions, but witha stride greater than 1, which causes the output feature maps to have smaller spatial dimensions than the input feature maps.
5) *Flatten:* The output of the last convolutional layer is flattened into a 1D vector, which is then fed into a series of fully connected layers. This flattening operation allows the output of the convolutional layers to be processed by the fullyconnected layers
6) *Fully Connected Layers:* The flattened vector is passedthrough a few fully connected layers, which are responsible for making the final prediction. The number of neurons in thefully connected layers typically decreases as we go deeper into the network.
7) *Sigmoid Activation Function:* The output of the last fully connected layer is fed into a sigmoid activation function, which produces a scalar value between 0 and 1. This value represents the probability of the input image being real.
8) *Output:* The output of the sigmoid activation function is the final output of the discriminator neural network, and represents the probability of the input image being real. During training, the goal is to maximize theouput for real images and minimize it for fake images generated by the generator network.

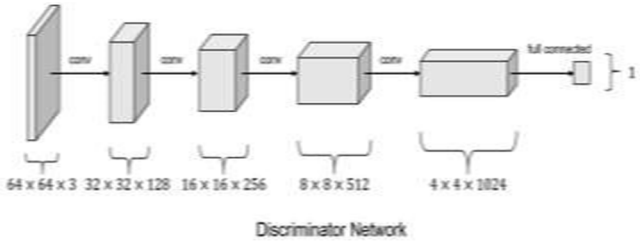


Figure7: Discriminator Architecture

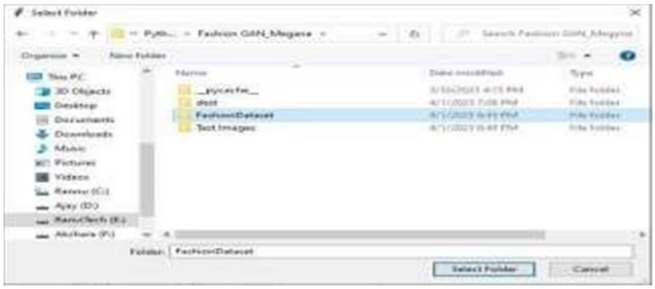
V. RESULTS

1) Home Page



Screen1: Home Page

2) Upload Dataset


Screen2: Upload Dataset

3) Enter Number option of Images to Generate



Screen3: Enter no. of images to generate

4) Enter any Number to Generate images



Screen4: Enter a number from keyboard

5) Generated 15 Images



Screen5: Generated images

6) Different colors of selected outfit



Screen6: Different colors of selected outfit

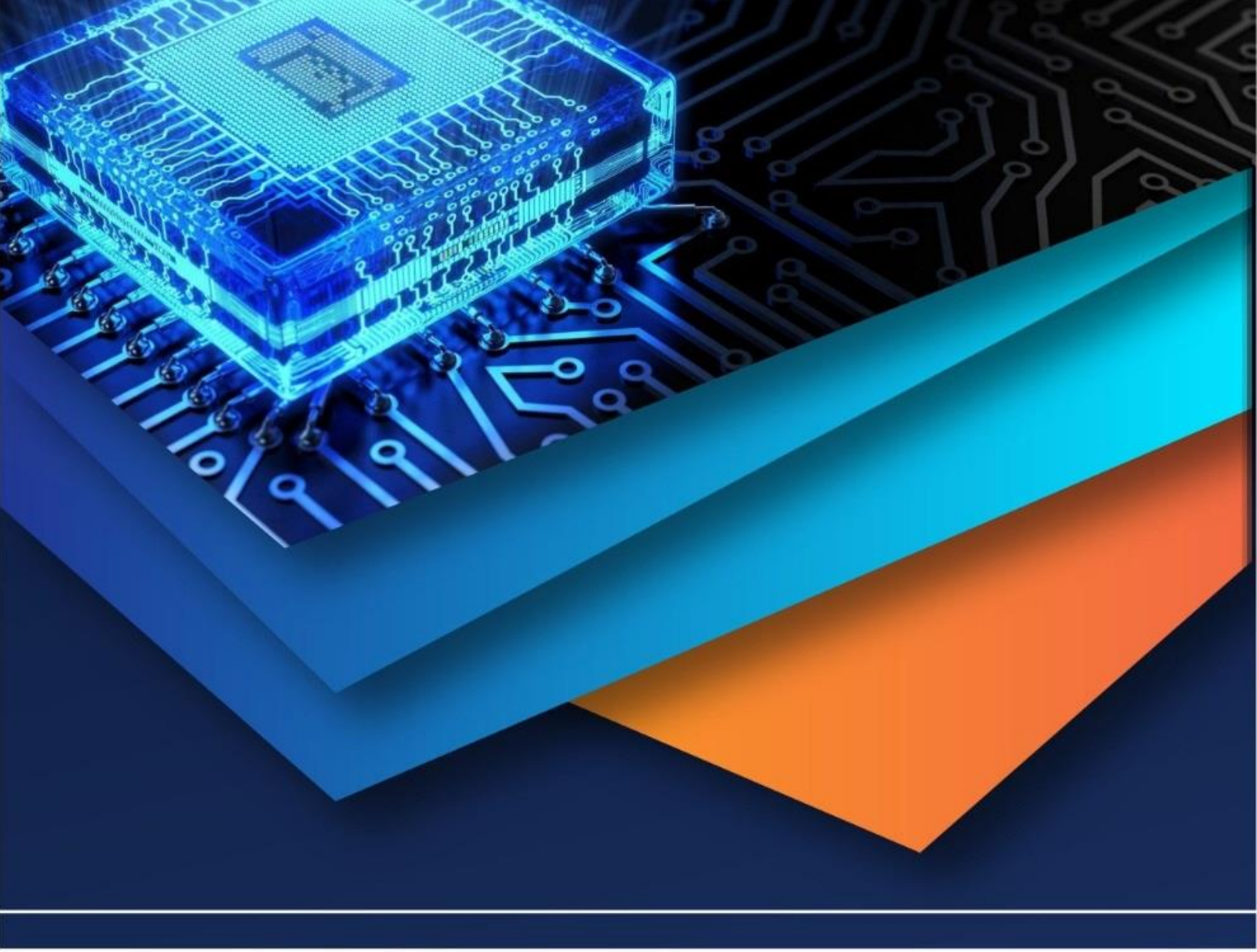
VI. CONCLUSION

In our model we used DC GAN – Deep convolutional GAN, through which we could generate fashion cloth images by using a generator model and allowing the user to save the obtained output in a user specified location along with a choice to choose colors of the generated output by using image processing (image color translation techniques).

REFERENCES

[1] AN Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, et al. Generative adversarial nets. In NIPS, 2014.
[2] Mily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. arXiv preprint arXiv:1506.05751, 2015.
[3] C. M. Bishop, M. Svens´en, and C. K. I. Williams. GTM: The generative topographic mapping. Neural Computation, 10(1):215–234, 1998.
[4] J. Gauthier. Conditional generative adversarial nets for convolutional face generation. Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester 2014, 2014.
[5] G. K. Dziugaite, D. M. Roy, and Z. Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In UAI, pages 258– 267, 2015.
[6] Graves. Generating sequences with recurrent neural networks. arXiv:1308.0850, 2013.
[7] Isola, P., Zhu, J.Y., Zhou, T. and Efros, A.A. (2017),“Image-to-image translation with conditional adversarial networks”,Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,Honolulu, HI, pp. 1125-1134.
[8] Lassner, C., Pons-Moll, G. and Gehler, P.V. (2017), “A generative model of people in clothing”, Proceedings of the IEEE International Conference on Computer Vision,Venice, pp. 853-862.
[9] Xian, W., Sangkloy, P., Agrawal, V., Raj, A., Lu, J., Fang, C., Yu, F. and Hays, J. (2018), “Texturegan controlling deep image synthesis with texture patches”,Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,Salt Lake City, UT, pp. 8456-8465.
[10] T. Cooper, “Color Segmentation as an Aid to White Balancing for Digital Still Cameras”,Proc. SPIE,Vol.4300, pp. 164-171,200
[11] Alec Radford, Luke Metz, Soumith Chintala "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks", 19Nov 2015
[12] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Sepp Hochreiter "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium", NeurIPS2017
[13] "Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab" by Chris Solomon and Toby Breckon,20 December 2010.
[14] Wayne Niblack, An Introduction to Digital Image Processing, Prentice-Hall International, 1985. ISBN 0-13-480674-3
[15] William Pratt, Digital Image Processing, (Third Ed) Wiley-Interscience, 2001. ISBN 0-471-37407- 5
[16] Advanced Deep Learning with Keras: Apply deep learning techniques, autoencoders, GANs, variational autoencoders, deep reinforcement learning, policy gradients, and moreby Rowel Atienza

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)