Creating Objects for Metaverse using GANs and Autoencoders

Aditya Singh[1], Tejas Patil[1]

[1]*Student, Dept. of Computer Engineering, Datta Meghe College of Engineering, Navi Mumbai, Maharashtra, India*

Abstract – *Metaverse refers to online spaces which allow people to interact with one other in a more captivating way than a website. This can be achieved through virtual reality (VR) and augmented reality (AR). To make the Metaverse more connected to reality, we make use of real-life objects that translate into the Metaverse, that's where the need to have state of an art model to achieve this goal. In this paper, we have proposed a way to create such objects using Generative Adversarial Networks (GANs) coupled with an Autoencoder. GANs will create a new object, the object could be anything from animals to human faces, cartoon figures etc. With the creation of such objects, the goal is basically achieved here. However, the output given by the GAN model is not of very high-quality that's where Autoencoder comes in with the use of Autoencoder the object is upscaled (super-resolution).*

Key Words: **Metaverse, Generative Adversarial Networks (GANs), Autoencoder**

1. INTRODUCTION

Metaverse [1] is like ad immersive world where the user can interact with others, attend a concert etc. which is achieved with AR and VR technologies. In this VR world, we may make our model and react with such different models this paper aims to make this goal more realistic by creating objects based on real-life entities with which these models (people) can react.

1.1 Generative Adversarial Networks (GANs)

Generative Adversarial Networks [2] also known as GANs consist of a pair of neural networks namely generator and discriminator that compete with each other. As the name suggests, the generator has to generate images while the discriminator has to detect whether the image generated is real or fake. The discriminator is fed at random real images from the data set it's trained on and images generated by the generator, it has to successfully identify the real image. On the other hand, the generator has to generate a better-quality image such that the discriminator fails to discriminate between the real image and generated image. The basic idea of a GAN model is to create a fake never seen image with help of existing data. The model stops training when the Nash equilibrium is achieved.

The following is the general loss function of GAN called Minimax loss there are many other variations of this loss function.

$$E_x[log(D(x))] + E_z[log(1-D(G(z)))]$$

In the above minimax loss equation, D(x) is the discriminator's estimate of the probability that the present data is real. E_x is the expected value over all real data instances. G(z) is the output given by the generator from noise G. D(G(z)) is the estimate of the probability that the fake instance is real

1.2 Autoencoders

Autoencoder [3] is an interesting variant of a Fully connected neural network, It consists of three parts the encoder, the bottleneck and the decoder. The encoder takes the input and its compression is performed in order to store the spatial data in the bottleneck which only consists of three neurons, Therefore the bottleneck stores the spatial data in a lower dimension. this data is further fed into the decoder which is again a fully connected layer and the original image is generated. The bottleneck is forced to learn important information in order to compress the data into lower dimensions such that using the same data the original image can be generated. Autoencoder just like GAN has many variations depending upon the fully connected layer sometimes it's replaced by pooling layers. Depending upon the variation of Autoencoder the loss function may vary, mentioned below is general a loss function:

$$min\ E(A, B) = min\ \Sigma E(x_t) = min\ \Sigma\Delta(A \circ B(x_t), x_t)$$

2. PROPOSED WORK

In this paper, we propose a model which will generate real-world objects including human faces with help of a Generative Adversarial Network model and upscale the pixel quality of the generated object with help of Autoencoders, this generated object will be further used in AR/VR application to serve the need of brining real-life feel in Metaverse. For the GAN model, we made use of the Deep Convolutional Generative Adversarial Network (DCGAN) [4] which consists of the deep convolutional layer in the generator and discriminator. In the case of the autoencoder, we made use of both the convolutional layer and pooling layer for both encoder and decoder. the GAN model and Autoencoder model were integrated and the output generated was ready to be fed into the AR/VR application. In the next section, we will look into this process in more detail as we explore the methodology

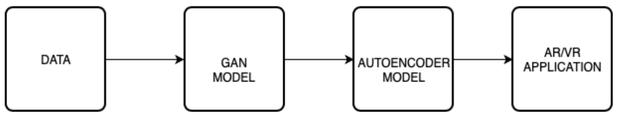

Fig -1: Structure of the Application

3. METHODOLOGY

3.1 Generating images using GANs

A custom dataset was created by us which consisted of human faces with more than 5000 images. the images were preprocessed reshaped in 28x28x1 and fed into the generator which consisted of a dense layer and three convolutional layers, Batch normalization [5] was used and leaky ReLU [6] as the activation function. In the case of the discriminator three convolutional were used and in the final layer, the sigmoid activation function was used. For the optimizer, we chose the Adam optimizer to train the discriminator.

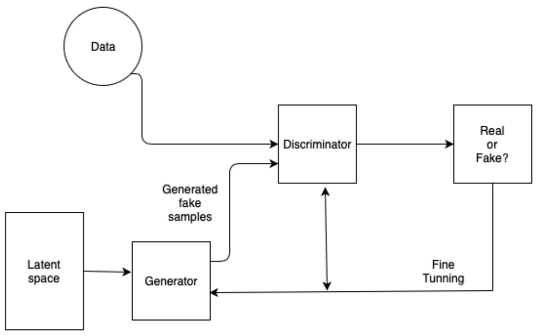

Fig -1: Architecture of DCGAN Model

3.2 Super-image resolution using Autoencoders

The output generated from the DCGAN model is fed into the autoencoder model. The autoencoder model consists of three parts the encoder, the decoder and the bottleneck which learns the latent space representation of the data. For the encoder, we made use of 7 which consisted of convolutional layers, Dropout layers and Pooling layers. Meanwhile, for the decoder, we used 15 layers consisting of convolutional layers, pooling layers and upscaling layers. We made use of ReLU activation and L1 normalization. The training was done in batches.

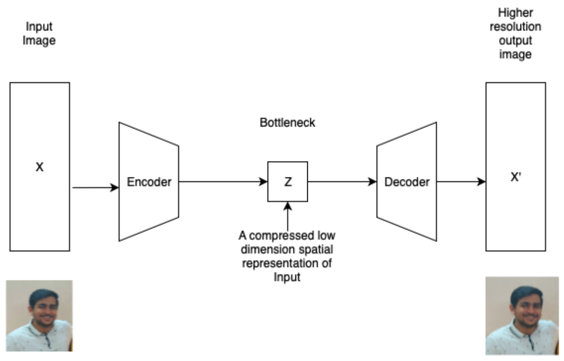
Input Image
Higher resolution output image
X
Encoder
Bottleneck
Z
Decoder
X'
A compressed low dimension spatial representation of Input

Fig -2: Output of the Autoencoder Model

4. RESULTS

4.1 Result from the DCGAN Model

The following are outputs produced by the Deep Convolutional Generative Adversarial Model (DCGAN). Based on the 5000 image data the model was trained on.

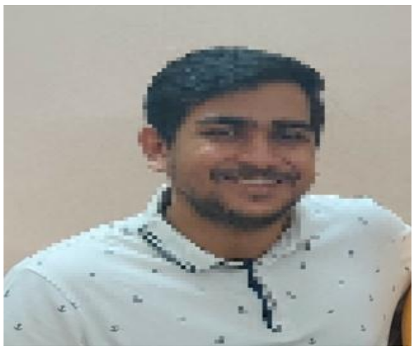

Fig -1: First result generated by the DCGAN model

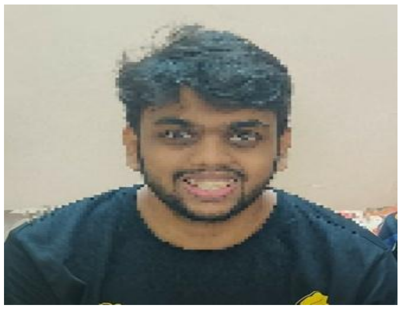

Fig -2: Second Result generated by the DCGAN model

4.2 Super Image Resolution using Autoencoders

After getting outputs from the DCGAN model the results were fed into the Autoencoder model below are super image resolutions of the outputs ready to be fed into AR/VR applications

Fig -3: First Result generated by the Autoencoder model

Fig -4: Second Result generated by the Autoencoder model

5. CONCLUSION

In this paper, we have proposed a way to make the metaverse more connected to the real world by including human faces the approach can however be even used to include other objects like cars, animals etc. The Deep Convolutional Generative Adversarial network was responsible for generating the images and Autoencoders was used for image supper resolution to upscale the images. The images generated can be used in AR/VR applications.

REFERENCES

1) Stylianos Mystakidis, "Metaverse", Encyclopedia, https://doi.org/10.3390/encyclopedia2010031

2) Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, "Generative Adversarial Networks", arXiv:1406.2661v1

3) Dor Bank, Noam Koenigestein, Raja Giryes, "Autoencoders", arXiv:2003.05991v2

4) Alec Radford, Luke Metz, Soumith Chintala, "Unsupervised Representation Learning with Deep Convolitional Generative Adversarial Networks", arXiv:1511.06434v2

5) Segey Ioffe, Christian Szegedy, "Accelerating Deep Network Training by Reducing Internal Covariate Shift", https://doi.org/10.48550/arXiv.1502.03167

6) Bing Xu, Naiyan Wang, Tianqi Chen, Mu Li, "Empirical Evaluation of Rectified Activation in Convolutional Network", arXiv:1505.00853v2