International Journal of Multidisciplinary and Current Educational Research (IJMCER)
ISSN: 2581-7027 ||Volume|| 4 ||Issue|| 3 ||Pages 150-155 ||2022||

Human Face Generation Using DCGAN

[1,]Piyusha Babar, [2,]YashanjaliChavan, [3,]ShitalIndulwar, [4,]Dr. Sharada Ohatkar*

Department of Electronics and Telecommunication,
MKSSS's Cummins College of Engineering for Women, Pune, India

ABSTRACT: In the past few years, Artificial Intelligence has been flourishing in numerous computer vision applications. In this project, we define and train a model with an advanced class of CNNs called Deep Convolutional Generative Adversarial Network (DCGAN) on a dataset of faces. The main objective of the model is to get a Generator Network to generate new images of human faces from the given dataset of images that look as realistic as possible to the original. This technology has resulted in an evolution in the animation and advertising industries. Acquiring and processing the data set for the Machine learning technology is one of the time-consuming processes, so Generative Adversarial Neural Network (GAN) is introduced. GAN is known to be one of the most promising methods of unsupervised learning in recent years and typically works with image datasets. We explore the potential of GAN to generate realistic images.

KEYWORDS: Convolutional Neural Networks, Deep Convolutional Generative Adversarial Network, CelebA

I. INTRODUCTION

A translation from one domain to another is the process of learning the map between an input image and an output image, where the goal is to create a mapping between an input image and an output image. Creating such images, especially with the help of a manual method is a laborious task in obtaining better results. Applying such alterations to images such as imposing a new face is a cumbersome task. Fake faces/ Deep Fakes find use in a variety of applications such as education, entertainment industry, fashion industry, etc. Computer vision studies have recently been intrigued by deep generative models for synthesizing realistic images. The utilization of Generative Adversarial Networks (GAN) for generating new images has shown impressive results in recent machine learning research. They have been studied extensively since 2016, and different variants exist that apply to image processing, computer vision, music, speech, audio, and data science. In unsupervised data, a specific GAN termed the Deep Convolutional Generative Adversarial Network (DCGAN) is used. The Generator and Discriminator deep artificial neural networks are combined in the DCGAN. When it comes to photos, DCGAN finds that its scope is expanding. Filling neighboring pixels to erase a subject from a picture or to eradicate a bad effect like red-eye or a watermark is called inpainting. However, there is more to learn about outpainting, and new strategies must be implemented. The main purpose of outpainting is to develop tons of new pixels in order to create an image that seems equivalent to the source. DCGAN aids in the iterative outpainting of the image, resulting in larger extended realistic images. We have trained Deep Convolutional GAN (DCGAN) using a collection of portraits of celebrities. By mapping random noise into images, the Generator network makes it impossible to determine which images came from the dataset and which images came from the generator.

II. LITERATURE SURVEY

[2]The research reported in this publication described forensics models that were demonstrated to be generalizable to various CNN-synthesis methods after being trained on CNN-generated images. For recognizing CNN-generated images, a new dataset and measure were also developed. According to the paper, a universal detector may be used to distinguish today's CNN-generated images from genuine photos, regardless of the architecture or dataset employed.[1]The researchers were able to strike a balance between supervised and unsupervised learning for CNNs with their study. Deep convolutional generative adversarial networks (DCGANs), a new class of CNNs with architectural limitations, have been introduced which makes them robust to train in most scenarios. They have applied the trained discriminators for image classifiers, exhibiting superior results with existing unsupervised methods. They show empirically that certain filters have learned to sketch things by visualizing the filters learned by GANs and also have demonstrated that the generators have intriguing vector mathematical properties that make altering various semantic aspects of generated samples simple. Unsupervised learning was proved to be a good fit for these networks.

This research presents more steady architectures for training GANs and provides proof that adversarial networks for supervised learning and generative modeling learn good representations of pictures. [3]GANs have proved to be a promising technology in the field of machine learning as they have filled the setbacks created by Self Organized Networks which caused concern about the availability of real labeled data. The following work examines GANs and their methods on a deeper level. GANs and their applications have successfully resulted in converting low-resolution images to higher-resolution images and producing convincing solutions. It uses LeakyReLU as an activation function in two deep neural networks to get smooth and efficient images. DCGAN which is a combination of a generator and discriminator network finds its purpose in unsupervised data. Outpainting is the need to produce new pixels similar to the original image. DCGAN is known to outpaint the images recursively to a greater extent. Training the image dataset for a small number of epochs can result in only limited output efficiency hence by increasing the number of epochs, optimizing neural layers and learning rates better and efficient results can be obtained. With all said, despite the advancements in GANs it still lacks proper metrics and unstable training.

[4] GANs might be a major advancement in generative modeling but they are also equally challenging to train. Instead of exhibiting long-term stability, the loss between the Discriminator and the Generator might start to oscillate frantically. A small amount of oscillation of the loss between batches is acceptable but typically in the long-term loss that stabilizes or moderately increases or decreases is favored to ensure that GAN converges and improves over time. When the Generator is successful in finding a small batch of samples that fool the Discriminator, Mode collapse is known to occur, and therefore the Discriminator is unable to produce any examples other than the limited set. Even if we manage to restrain the Discriminator to stop it from being fooled by this one particular point, the Generator will simply however find another mode as the Discriminator has become numb to its input. Another issue faced while training the GANs is the lack of correlation between image quality and Generator loss. Even with simple GANs, a large set of hyperparameters for both architectures are to be tuned. GANs are highly sensitive to such changes and hence it is important to understand their inner workings in order to find the right set of parameters that might improve the stability and efficiency of the model. However, advancements have been made in recent years to tackle some of the issues mentioned earlier and improve stability.[5] The researchers were successfully able to propose a framework for estimating generative models through an adversarial process, in which two models are y trained at the same time. The Generator model is trained for capturing the distribution of the data whereas the Discriminator model is supposed to estimate the probability. However, the Discriminator must be synchronized well with the Generator model during training in order to avoid the Generator from collapsing from too many values of the latent noise. Nevertheless, no interference is required while training the model and also a variety of functions can be incorporated as well while also representing very sharp distributions. This is known to improve efficiency while also accelerating the training.

III. METHODOLOGY

In this section, we will see the architecture and mechanism behind the DCGAN model. This section describes the working of the generator and discriminator in DCGAN.

3.1 What are GANs?

GANs are producing models; they generate new fake human faces or images that are similar to that of the input dataset. There are two neural networks present in the GANs, the first is a generator and the second is a discriminator. Both are competitors of each other, competing to be highly precise in their respective forecasts. The generator aims to grasp an arrangement similar to the input dataset to generate lifelike or realistic human faces at the output. Random noise is given to the generator network as an input. The generator runs the noise throughout the neural network like the differentiation function to process the noise and reconstruct it to have life-like images present in the input or training dataset. The output of the generator network completely varies according to the selection of random input noise. Then comes the discriminator network, as the name (discriminator) suggests, it works as a classifier. The Discriminator network has two input datasets, generated images by the generator and the actual training dataset. It tries to classify the real images (existing human faces) and fake images. Discriminator assigns probabilities ranging from 0 to 1, for fake and real images.

3.2 Math behind GAN

Discriminator uses binary classification to classify real images from fake or generated images. This loss function is simply a Binary Cross-Entropy and trying to be a more precise discriminator tries to maximize the Binary cross Entropy. The equation for binary cross-entropy loss of discriminator can be represented as,

$$L(D) = \max[\log(\ D(x)) + \log(1 - D(G(z)))] \quad \text{..... (1)}$$

As mentioned above generator is a competitor of the discriminator, it will try to minimize the binary cross-entropy loss and the equation for the is exactly opposite to the Eq (1),

$$L(G) = \min[\log(\ D(x)) + \log(1 - D(G(z)))] \quad \text{..... (2)}$$

If we combine both the equations i.e. Eq (1) &Eq (2) results will be,

$$\min_G \max_D V(D,G) = \min_G \max_D (E_{x \sim P_{data}(x)}[\log D(x)] + E_{z \sim P_z(z)}[\log(1 - D(G(z)))]) \quad \text{…(3)}$$

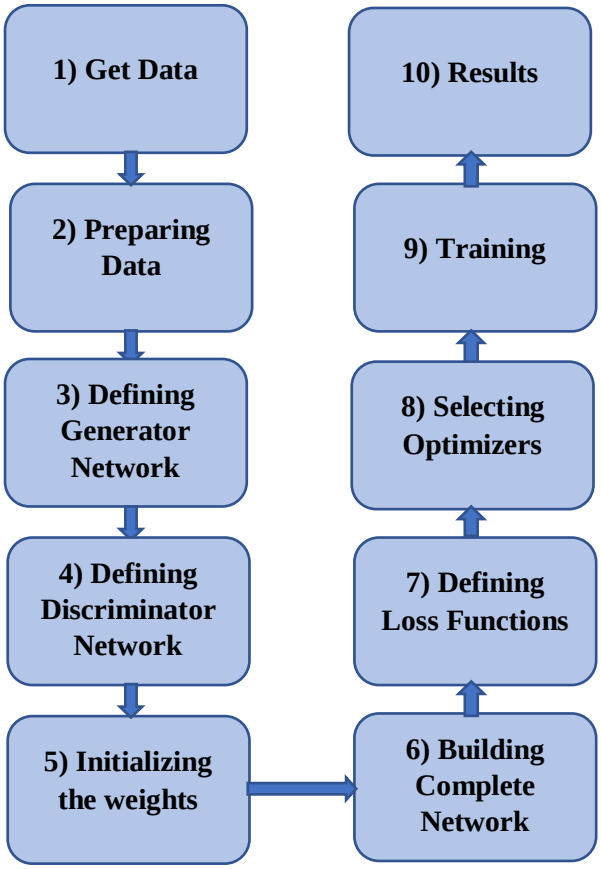
From Eq (1) &Eq (2),

$$L(G) = -L(D) \quad \text{..... (4)}$$

The discriminator and generator work in opposition to each other. The generator would want to maximize the loss functions and gives probability near to one or one for real data, whereas the discriminator would want to minimize the loss functions and give probability near to zero or zero for fake images.

IV. DESIGNING MODEL

In this section, we will see the detailed working of the model



4.1Getting the Data:
The adversarial network is trained with the CelebFaces Attributes Dataset (CelebA). The CelebA dataset is a miscellaneous dataset having around 250k images which are necessary for better training of the network.

4.2) Preparing Data:
The main objective was to build a model, so we have used preprocessed data.

4.3) Defining Generator Network:
Because our data is in the form of photos, transforming z to latent space entails constructing an RGB image of the identical dimension as the images used for training. This is achieved in practice by using a succession of strided 2D convolutional transpose layers. Except for the last one, each transpose convolution layer is followed by a Batch Normalization. We implemented the ReLU activation function to activate the concealed units. The height and width of each transpose convolution layer are doubled. For instance, 2x2 photos will be scaled into 4x4 after the first transpose convolution, so on and so forth. The generator output is sent via a tanh function to restore it to the [-1,1] input image data range. The presence of batch normalization functions after the convolution-transpose layers is noteworthy, as this is a key contribution of the DCGAN study. These layers aid in the flow of gradients during training. This architecture is designed so that the output of the fourth transpose convolution layer produces a 64x64x3 image.

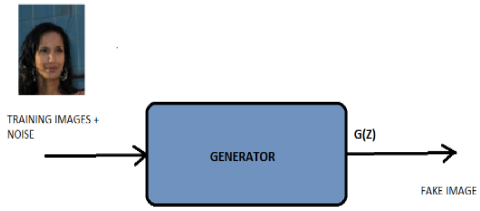


Fig.1: Generator representation

4.4) Defining Discriminator Network:
Discriminator accepts a 3x64x64 image as input, runs it through a succession of Conv2d, BatchNorm2d, and LeakyReLU layers, then uses a Sigmoid activation function to produce the final probability. Excluding the first, every convolution layer is followed by a Batch Normalization layer. The final completely connected layer outputs a single logit to describe if the image is real or not. As explained in the methodology section, we have used Leaky ReLU activation functions for the hidden units. If more layers are required for the problem, this architecture can be expanded, but the employment of strided convolution, BatchNorm, and LeakyReLUs is important. Using strided convolution rather than pooling to downsample is recommended by the DCGAN paper because it allows the network to train its own pooling function. Furthermore, the batch norm and LeakyReLUfunctions support a healthy gradient flow, which is essential for both Generator and Discriminator learning processes.

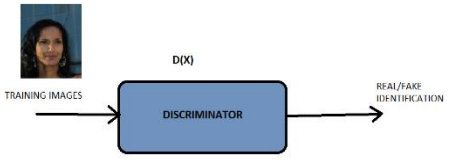


Fig.2: Discriminator Representation

4.5) Initializing the weights:
- According to the authors of the DCGAN publication, all model weights must be randomly initialized from a Normal distribution with mean=0 and standard deviation=0.02. To achieve these requirements, the weights_init function takes an initialized model and resets every convolutional, convolutional-transpose, and batch normalization layer. This function is applied to the models just after they are created.

4.6) Building Complete Network:
-The model's hyperparameters are defined and instantiation of the discriminator and generator is done.

4.7) Defining Loss Functions:
-We can determine how they learn using the loss functions with D and G configuration. The Binary Cross-Entropy Loss function will be used. It's worth noting that this function calculates both log components in the function. With the y input, we can specify which section of the BCE equation to employ. This will be done in the training loop, but it's vital to know how we can select whichever component we want to calculate simply by adjusting y. After that, we set the true label to 1 and the phony label to 0. When computing the losses of discriminator and Generator, these labels will be applied, as well.

4.8) Selecting Optimizers:
-It is intended to run both algorithms simultaneously to keep improving both networks. Two optimizers are defined for GANs, one for the Generator and one for the Discriminator.In both cases, Adam optimizer has been used.

4.9) Training:
-As part of the training process, the discriminator and generator have been alternately trained.The generator should then try to fool the discriminator by having an opposite loss function.

V. SIMULATION RESULTS

A few of the project's results are mentioned in this section, which can be used to gauge the model's performance.

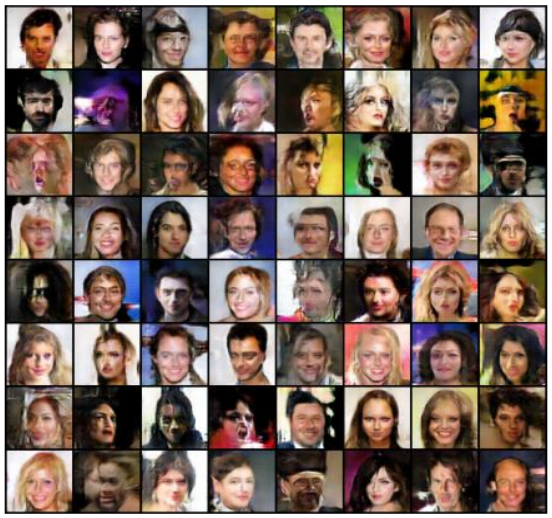


Fig.4: No. of Images- 244,000 and No. of Epochs- 5
Repetitive noise patterns over numerous samples appear to indicate perceived under-fitting.

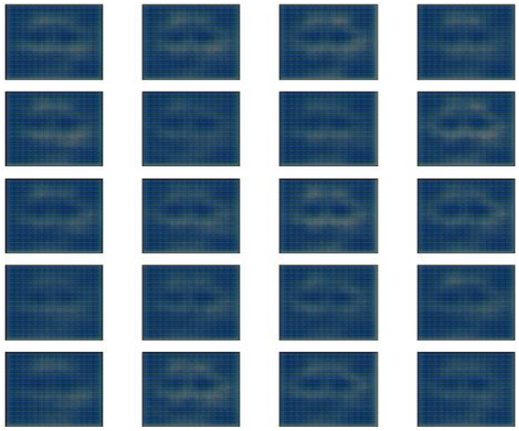


Fig 5: No. of Images- 100 and No. of Epoch- 50
The result obtained after input of 100 images is a blurred outline of a face.

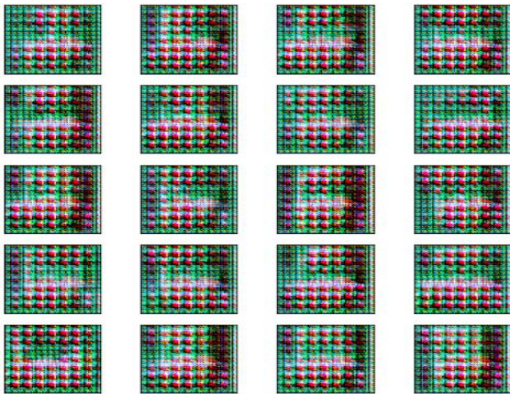


Fig 6: No. of Images- 1000 and No. of Epochs- 50

The result obtained after input of 1000 images is RGB images with lot of noise.

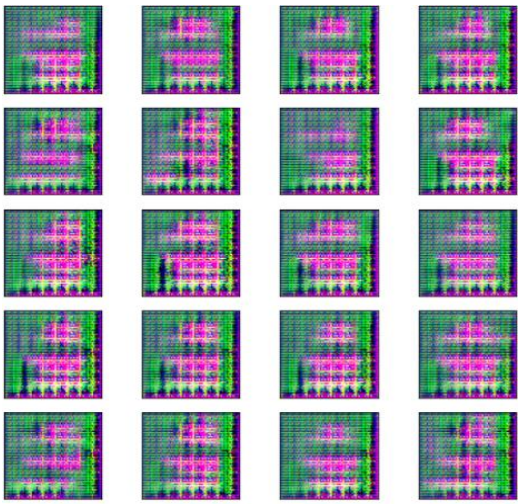


Fig 7 : No. of Images - 5000 and No. of Epochs – 50
The result obtained after input of 5000 images is a RGB image with a faint outline of a face and a lot of noise.

VI. CONCLUSION

We were capable of achieving the present outcome after only five epochs of training. Better performance could be achieved by increasing the amount of epochs and optimizing the learning rate and neural layers. This model will be efficient in unsupervised learning tasks and can obtain outcomes of the highest development with similar datasets by enhancing the specified network. One of the concerns with 'synthetic creation' is the possibility of losing the ability to discern what is genuine from what has been made artificially (so-called 'deep fakes'). We can't ignore the perils of using these technologies to classify people, but the reality remains that clever machines can discern the visual patterns that human expressions are based on.

VII. FUTURE SCOPE

To address this source of instability, more work is required. We believe it would be highly interesting to extend this approach to other fields including video (for frame forecasting) and audio (pre-trained characteristics for speech synthesis). It would also be interesting to look at the attributes of the learned latent space.

REFERENCES:

[1] Alec Radford & Luke Metz, SoumithChintala: Unsupervised Representation learning with deep convolutional generative adversarial networks: International conference on Learning representations, 2016.
[2] Sheng-Yu Wang1, Oliver Wang,Richard Zhang, Andrew Owens, Alexei A.:CNN-generated images are surprisingly easy to spot... for now:*IEEE*, 2020.
[3] Devaki P., Prasana Kumar, C.B Kaviraj S. and RamprasathA. :Face Generation using Deep Convolutional Generative Adversarial Neural Network: *Biosc.Biotech.Res.Comm. Special Issue Vol 13 No 11 (2020) Pp-20-23*, 2021.
[4] David Foster: Generative Deep Learning: *O'Reilly Media*, 2019.
[5] Ian J. Goodfellow, Jean Pouget-Abadie , Mehdi Mirza, Bing Xu, DavidWarde-Farley, SherjilOzair , Aaron Courville.: Generative Adverserial Networks YoshuaBengioDepartementd'informatique et de recherche op ´ erationnelle ´ Universite de Montreal ´ Montreal, *QC H3C 3J7*, 2014.
[6] Zhou Wang,Alan Conrad Bovik, Hamid Rahim Sheikh , Eero P. Simoncelli. :Image Quality Assessment: From Error Visibility to Structural Similarity: *IEEE*, 2004.