Deep Learning – Case Study

Fake Face Generator Using GAN.

Name: Parth Soni

Enrollment Number: 18012021081

Batch: DL2

1. Introduction

This case study is designed to fake face generator. It uses the GAN model to the generate images. Our objective is to create a model capable of generating realistic human images that do not exist in reality. (In future, I will upload a number of use cases on GAN and its variants). The technology behind these kinds of AI is called a GAN, or "Generative Adversarial Network". A GAN takes a different approach to learning than other types of neural networks(NN). GANs algorithmic architectures that use two neural networks called a Generator and a Discriminator, which "compete" against one another to create the desired result. The Generator's job is to create realistic looking fake images, while the Discriminator's job is to distinguish between real images and fake images. If both are functioning at high levels, the result is images that are seemingly identical real-life photos.

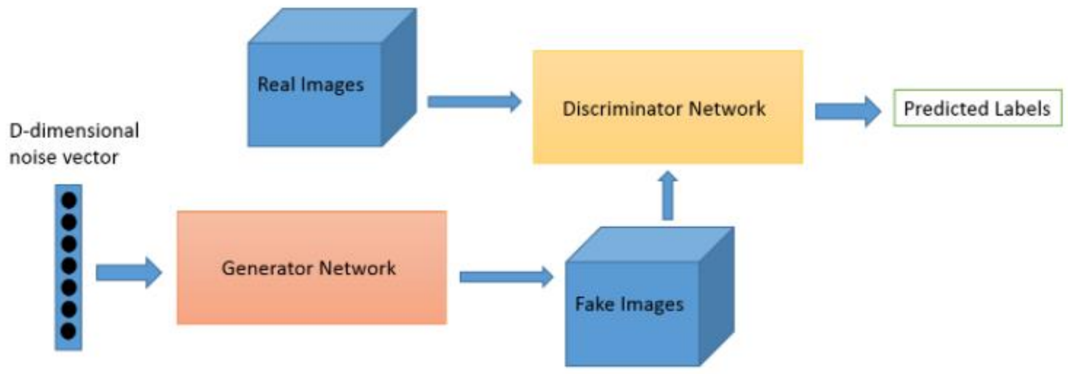
2. Tools and Technology

Tools and Libraries	Usage
Keras	This library is used for building the network architecture. It allows us to use several layers, callbacks, and InceptionResNetV2 model.
Tqdm	Provides a good progress bar that can be combined with a loop to get visualization of the current progress.
PIL	For loading the viewing the images.

3. Model Architecture

Generative Adversarial Networks (**GANs**), represent a shift in architecture design for deep neural networks. There are several advantages to using this architecture: it generalizes with limited data, conceives new scenes from small datasets, and makes simulated data look more realistic.

In GANs, there is a generator and a discriminator. The Generator generates fake samples of data(be it an image, audio, etc.) and tries to fool the Discriminator. The Discriminator, on the other hand, tries to distinguish between the real and fake samples. The Generator and the Discriminator are both Neural Networks and they both run in competition with each other in the training phase. The steps are repeated several times and in this, the Generator and Discriminator get better and better in their respective jobs after each repetition. The working can be visualized by the diagram given below:



Here, the generative model captures the distribution of data and is trained in such a manner that it tries to maximize the probability of the Discriminator in making a mistake. The Discriminator, on the other hand, is based on a model that estimates the probability that the sample that it got is received from the training data and not from the Generator.

The GANs are formulated as a minimax game, where the Discriminator is trying to minimize its reward V(D, G) and the Generator is trying to minimize the Discriminator's reward or in other words, maximize its loss. It can be mathematically described by the formula below:

$$\min_G \max_D V(D,G)$$

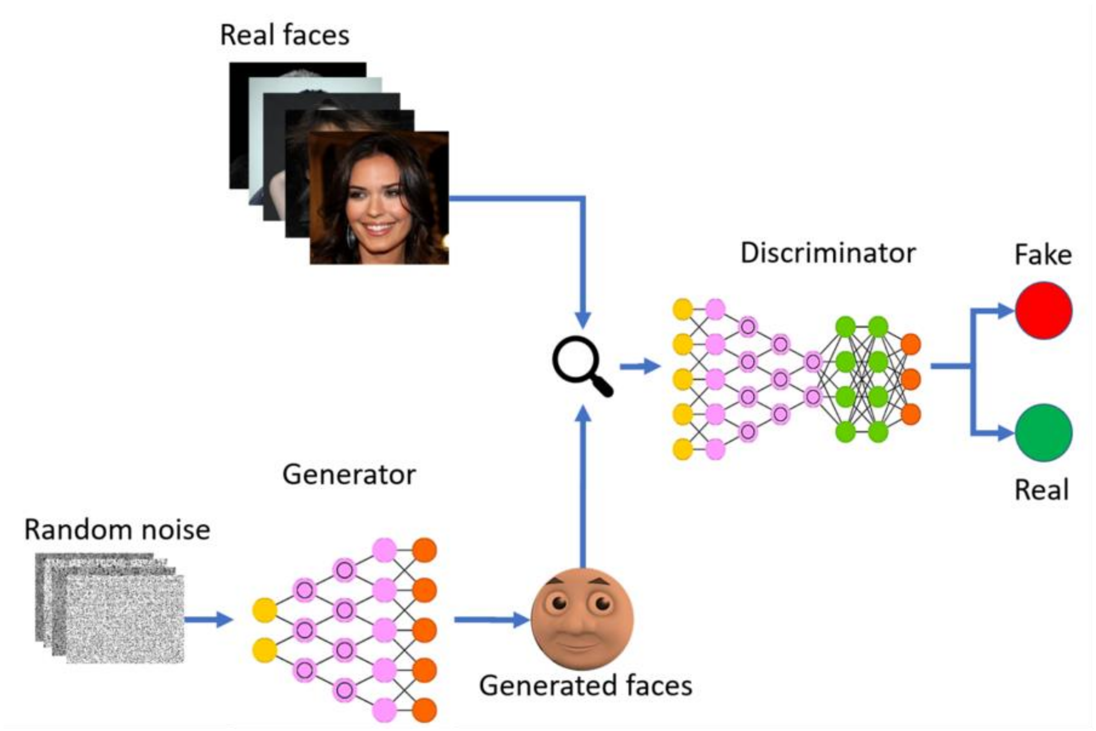
$$V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z))]$$

4. Working

Here I am using the real image to train the data set. This images will go in the generator to create another fake image with the help of real and compete with the created discriminator. So this will result in fake image.

Generated image will be predicted and after that it will be trained in discriminator then the GAN model will be trained. After Prediction of 50 images it will take backup automatically.

After All this process it will be save in Gif format including prediction of the fake image and so the graph of the prediction.



5. Code

https://github.com/parthsoni29/DL_CASE_STUDY/blob/main/Parth%20Soni_18012021081_DL2_Case%20Study.ipynb

6. Output