GENEARATIVE ADVERSARIAL NETWORK

Jenil Sanjaykumar Pandya 1017921730

Pratham Ramkripal Yadav 1017895120

GOOGLE DRIVE LINK : <https://drive.google.com/drive/folders/1Ee2XIuzXFbYEMvSt8xSmRrv716j4-VqH?usp=sharing>

Since the Files were too Large to Submit on Isidore here is a google drive link, Contact us If Any Issues with the Link.

Note: I was Initially using DCGAN to generate the landscape images although the code was working the output was not good enough (DCGAN Output shown in Figure 1).

Output with DCGAN

A close-up of a pixelated background

Description automatically generatedA green and black landscape

Description automatically generated with medium confidence A colorful pixelated background

Description automatically generated

*Figure 1*

**Some peers of** from my class helped me figure out a different approach to generate realistic landscape images. The Idea is to use Sketch to Color GAN to generate realistic images.

Approach:

Step 1: Convert the raw images of different width and height to a single consistent dimension of 256 \* 256 Pixels. I achieved the following by using the script named **Resize Images.ipynb.**

Step 2: Convert the Resized images into sketches so that it can be fed to the GAN network for training also merge the sketch with the original image. This will hep us in printing the input image i.e., the sketch image, the ground truth i.e., the original image and the Predicted Image one that is generated by the generator while training during the epochs. I achieved the following using a script named **Convert To Sketch.ipynb, because we** figured out in the class that all the landscape have Sun’s exposure ruining every other aspect of the picture, because of sun’s exposure we are not having good fake landscape, because sun’s glare is overpowering, so converting them to a sketch limits the sun exposure that was coming into training with Generator.

Step 3: Since all the files present in the dataset are in JPG format, I have written a code to convert all these JPG files into PNG file as my model accepts the input in PNG format.   
(I tried changing the format directly in the model but it was throwing compatibility issues thus decided to change the format of input data from jpg to png). I achieved the following using a script named **Convert JPG TO PNG.ipynb**

Step 4: Creating the Generator model of GAN Network. The Generator model is a replica of UNet Architecture that incorporates skip connections to other layers rather than going to immediate next layer. I also have incorporated up sampling and down sampling ideas as discussed in the last Help session.

Step 5: The discriminator model serves to distinguish between images generated by the generator model and those originating from the actual training dataset.

Step 6: Train the model. I have ran the above model for 10 epochs which took me approximately 3 hours to complete with each epoch running for an average of 15 minutes.

Final Output:

A close-up of a desert

Description automatically generated

*Figure 2*