CS512-Project-Report

Sai Manohar Vemuri | Raghunath Babu

A20514848|A20511598

Department of Computer Science

Illinois Institute of Technology

Quality Image Generation using Generative Adversarial Networks

Abstract:

This project presents improved results in generating quality images employing U-Net and Transfer learning architectures. The U-Net architecture helps in providing per pixel feedback and global coherence of the generated image. Along with this cut mix augmentation is appended in the discriminator model to improve the localization ability.

Problem Statement:

Existing GANs were a concatenation of several convolution layers resulted in expensive training which required a lot of computation. Also, in terms of the quality of the image generator, there was a need for global coherence to capture complex patterns and the overall structure of the input data with lesser computations.

Approach:

To get the per pixel feedback(local) and global coherence, an encoder decoder architecture U-Net is used. Along with this several transfer learning architectures such as inception for multi-receptive fields analysis and resnet(with residual block) in interest of generating quality images.

Augmentation is also included to improve the localization ability of the model and make the model immune to several transformations of the datasets.

Generative Adversarial Networks (GANS):

GANS is a generative modelling approach using convolutional neural networks or deep learning. It is an unsupervised approach which tries to learn the probabilistic distribution function of the input datasets (training data), in the larger picture it learns and hunts for patterns in the training data. It has 2 parts in it, the discriminator and the generator. The discriminator is the one that discriminates between two different classes of data whether the generated image is fake or not. Whereas the generator model is trained on the data that is sampled on a true distribution (training data). Given a standard normal distribution it produced on output as close to the distribution of the training data.

Say,

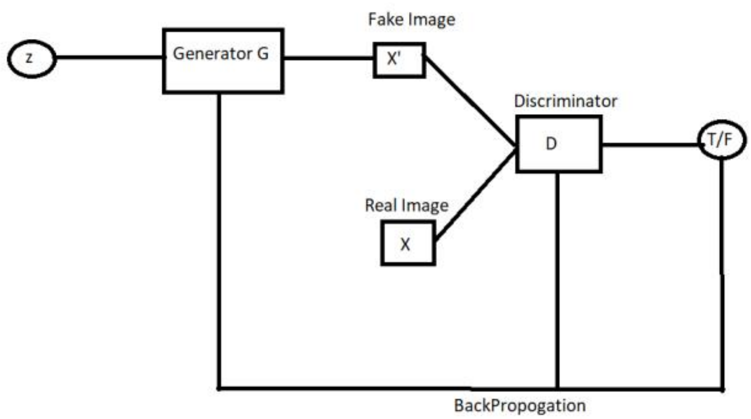
Generator model G

Standard normal distribution z

True distribution D

Generator models produce G(z)~D'

Back-propagation- during the training of the Generator the discriminators' weight and biases are fixed. This is done because discriminators should not be too strong to classify the images generated as real or fake, i.e. If the output is 0.5 the generator is fooled by the discriminator.



DataSet

FFHQ-Flickr-Faces-HQ is a dataset of human faces of 70000 images.

Data_set_FFHQ

Preprocessing

Initially a data set of 70,000 images were taken, from which a 1000 set of random images were considered for this project. These images are converted into NumPy arrays and then uploaded into google drive as a pickle file. This is done to ease the computational and reduce the time for loading and processing the data.

The proposed project is to use the U-Net based Generator.

U. Net Generator

In generative models, traditional convolutional layers requiring high computation lagged in quality image generation because of not having proper feedback from the generated image. This gave a scope of using an encoder-decoder architecture – U-Net.

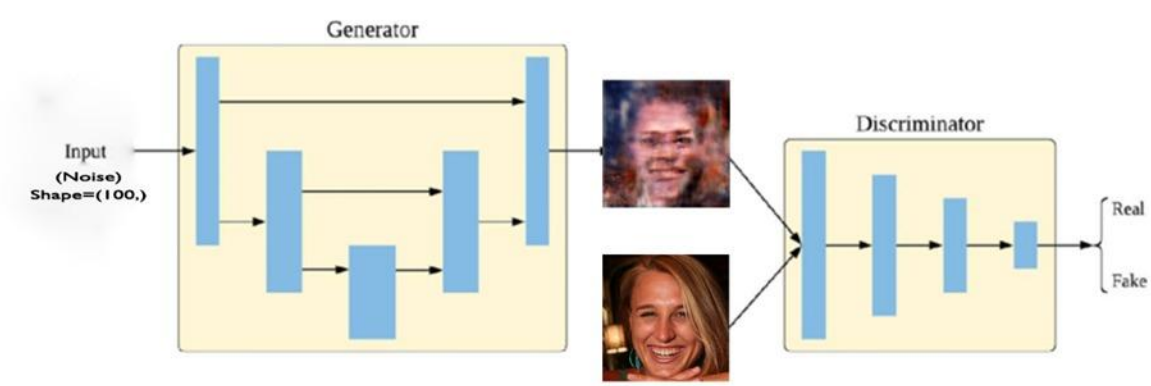
The encoder part in the U-Net performs per -pixel classification and the decoder part give per-pixel decision providing spatial coherent feedback to the generator. Overall, it concentrates on the semantic and structural changes between the real and the fake image, providing global and local coherence of the synthesized images. This type of generator resonated well for applications where there is need of mapping 2 domains without the need of paring training data such as cyclic GANS.

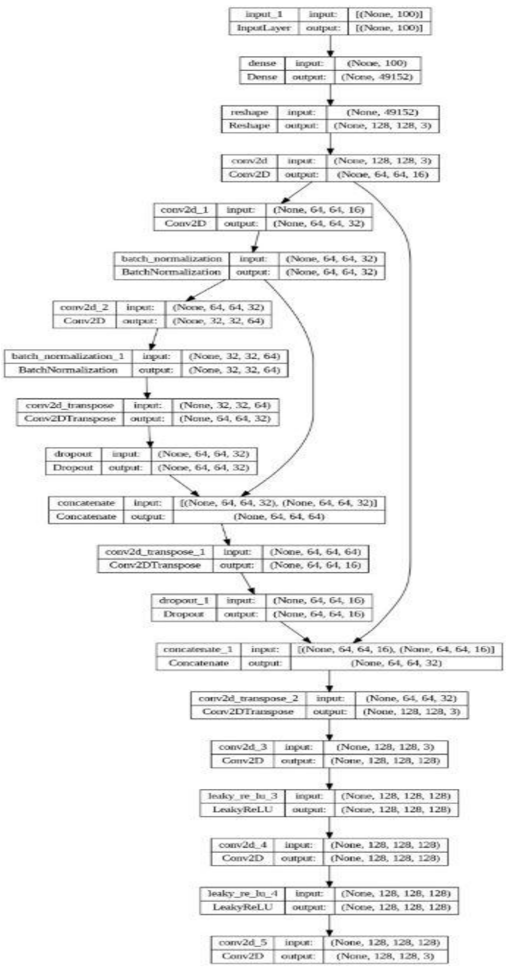
Architecture of Generator

Input image dimension is 224*224*3 is passed to the generator a U-Net architecture and a few convolutional layers. The encoder and decoder part consists of convolutional blocks with filters 32,64 and 64(middle layer) and the decoder part is unsampled as 64,32. Upon this a couple of convolutional layers with 128 filters are cascaded.

For each convolutional layer, Leaky Relu activation has been used.

The output of the generator results in dimensions the same as the input which is 128*128*3. For the output layer activation function used was tanh (1 to –1), because it helps in centering the generated images around mean zero and variance as close as the training samples.





Discriminator

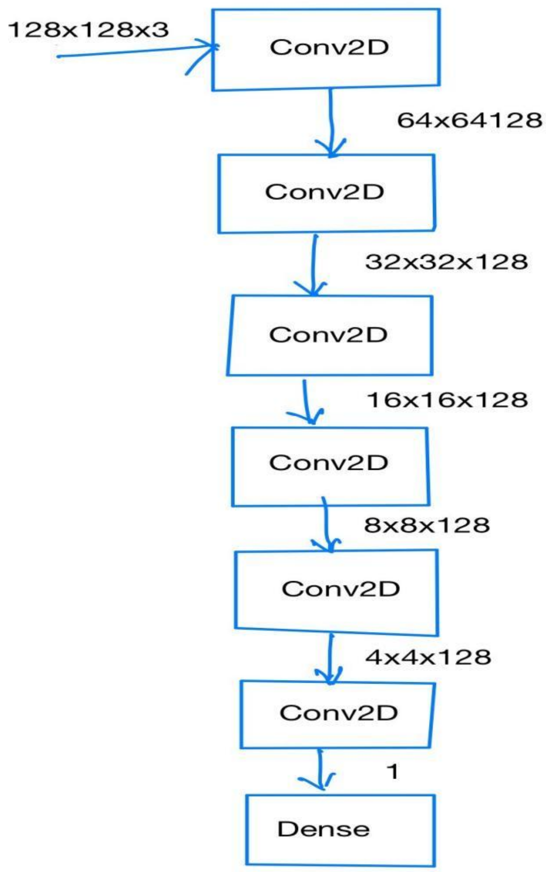
Architecture of Discriminator

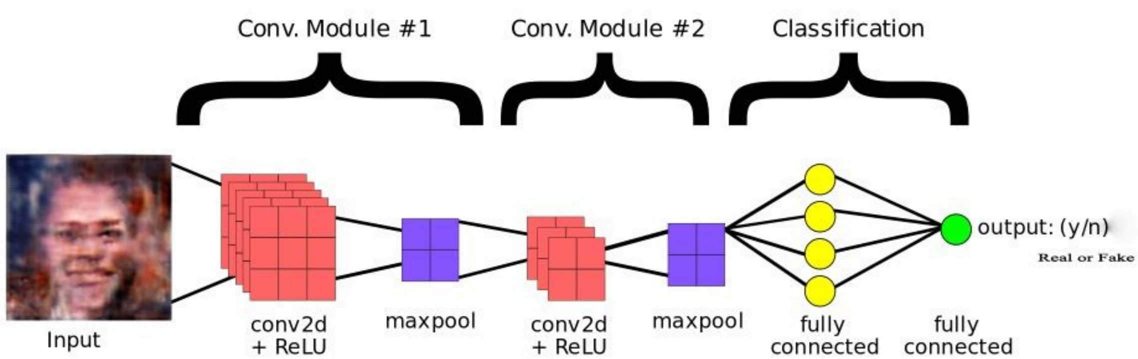
Input image of 128*128*3 is passed into the discriminator which is a combination of convolutional layer, each having activation function Leaky Relu, a dropout layer to prevent over fitting of probability 0.5. After passing this through 2 similar blocks this network is flattened to a binary output to assess the generated image as either fake or real.

Another model was built with cut mix-augmentation,

- Cut and mix patches from real and synthetic images.

Including this will make the discriminator focus on the semantic and structural changes between the real and the generated image, resulting in improved localization ability of the discriminator.





Loss

Regular GANs employs binary cross entropy

$L(y^\wedge,y) = [y*log(y^\wedge)+(1-y)*log(1-y^\wedge)$

For $y=1$

$y^\wedge=D(x)$

$L(D(x),1) = log(D(x))$

For $y=0$

$L(D(x),0) = (1-0)*log(1-D(G(z))$

$=log(1-D(G(z))$

Binary cross entropy with logit is the loss function used. The predicted probabilities of class 1 from the fully connected layer are sent to sigmoid activation function.

Using this combination of logit and binary cross entropy provides a probabilistic interpretation, enables smooth and differentiable optimization and supports balanced learning from imbalanced datasets.

Optimizer- Adam.

Learning rate

Generator – 1e-5

Discriminator –1e-4

Evaluation metrics

The evaluation metrics used to assess the quality of the generated image is SSIM and.

SSIM-Structural Similarity Index. It assesses the similarities between 2 images (real images and generated image) based on 3 components namely luminance, contrast and the structure resulting in values between –1 and 1.

$SSIM(x,y)=(I(x,y)*c(x,y)*s(x,y))$

FID: Frechet Inception Distance. Measures the degree of similarity of distribution between the generated image and the real image. (Lower the FID indicates closer the real images and the fake image).

Results and Discussion

Models are defined

1. Basic Gan(simple cascading of convolutional layers)

2. Unet based Generator

3. Unet generator with Cut-mix Augmentation based discriminator

1. Basic GANs

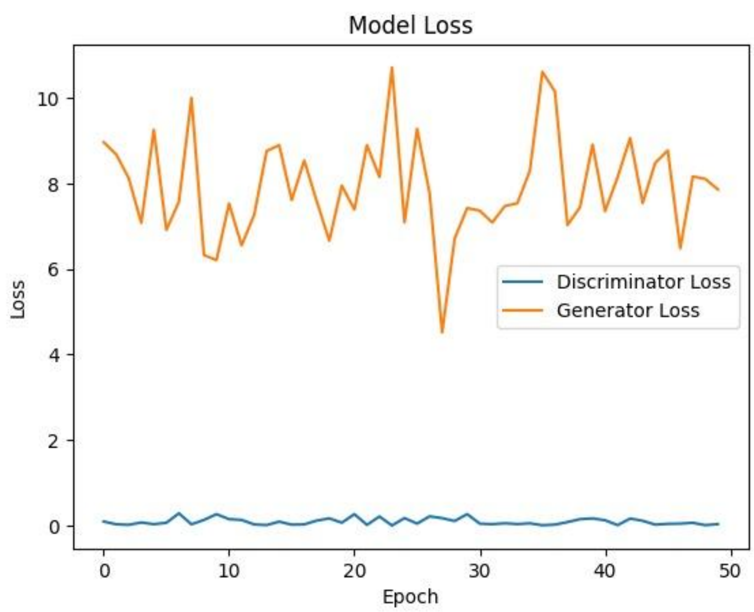
(with normal cascading of convolution(stride=2), batch normalization,Activation function-Leaky Relu

	SSIM	FID
Basic Gan	0.52511	8331

Final

Generator loss : 7.832

Discriminator loss: 0.075



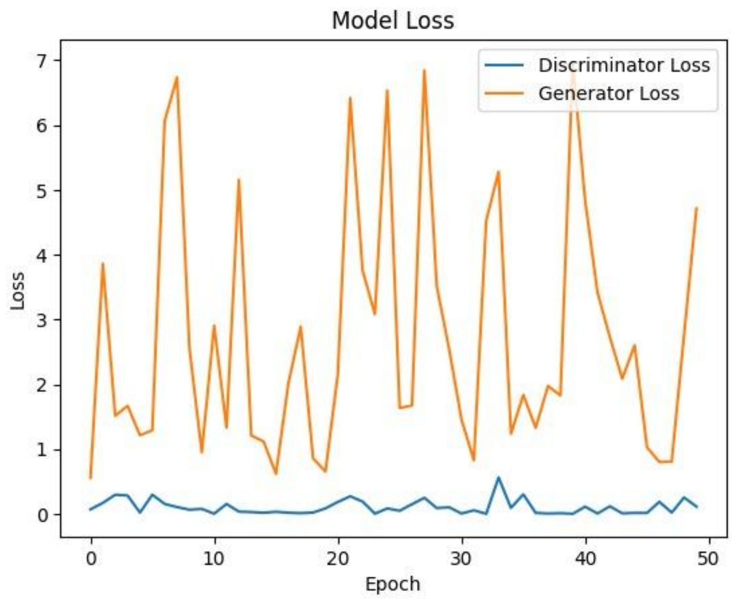
2. Unet based Generator

	SSIM	FID
Unet based Generator	0.58723	8231

Final

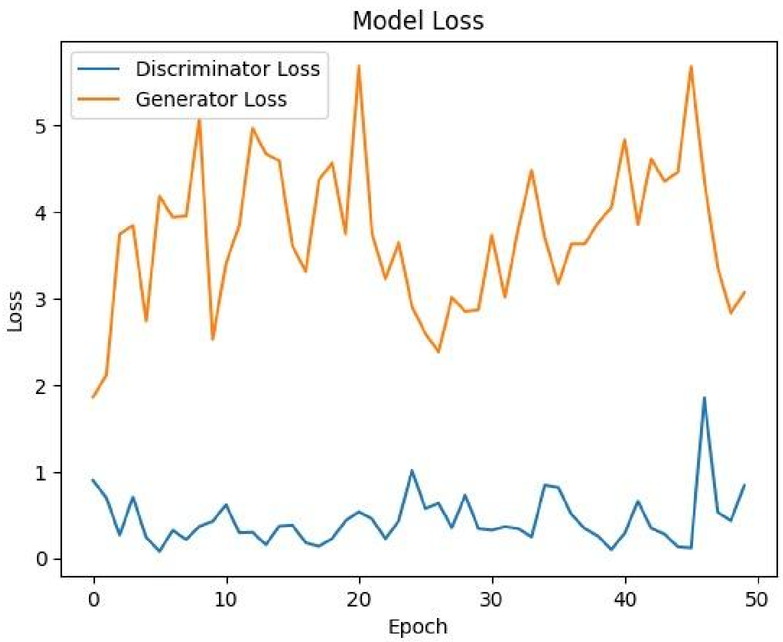
Generator loss : 3.0720

Discriminator loss: 0.0749



3. Unet with Cut-Mix Augmentation

	SSIM	FID
4. Cut-Mix Augmentation	0.5299	9474.4053



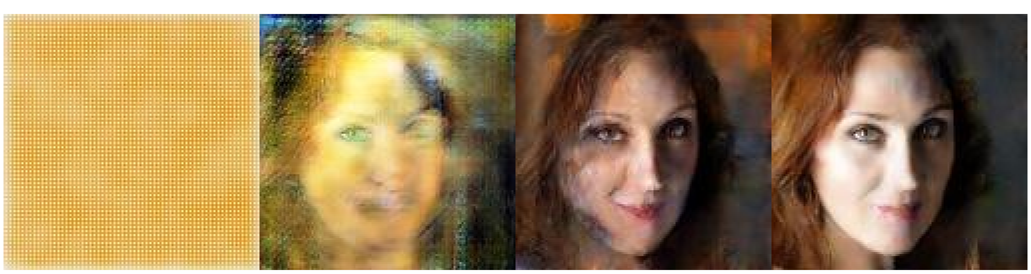
It is inferred that after several epochs the Basic Gan model image generation gets saturated, with high generator loss, which indicates that the generator is no longer able to fool the discriminator.

On the other hand, the U-Net based generator does not go to the saturation and still generates stable images with good accuracy compared to the basic Gan. This shows that the UNet based Generator is capable of delivering quality images, and if trained an addition of few hundred epochs it will give better results.

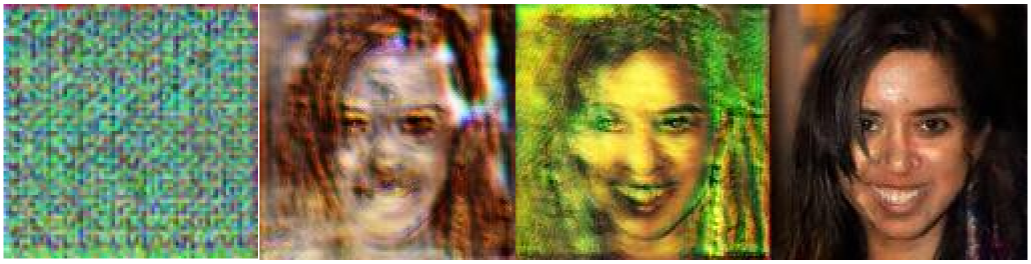
Our model has the potential to generate high quality images with minimal loss, despite trained on a shallow network due to limited computational resources. With more resources, our model could produce even better and become a powerful for creating quality images.

Output Images

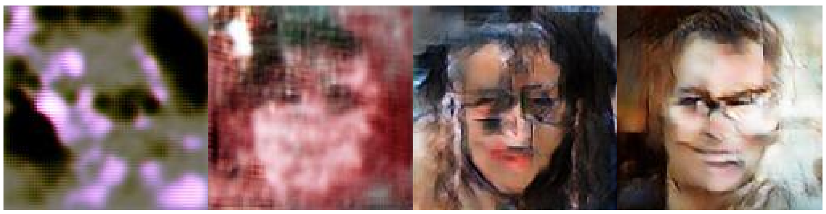
1. Basic GAN



2. UNet based GAN



3. CutMix Discriminator



Using UNet architecture with self attention(output 64*64)

Contribuion

	Student 1: Raghunath Babu	Student 2 : SaiManohar Vemuri
Data Preparation and	40	60

Literature Survey.		
Modelling	50	50
Evaluation & Report	60	40

References:

[1]. E. Schönfeld, B. Schiele and A. Khoreva, "A U-Net Based Discriminator for Generative Adversarial Networks," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 8204-8213, doi: 10.1109/CVPR42600.2020.00823.

[2]. Y. Wang, H. Laria, J. van de Weijer, L. Lopez-Fuentes and B. Raducanu, "TransferI2I: Transfer Learning for Image-to-Image Translation from Small Datasets," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021, pp. 13990-13999, doi: 10.1109/ICCV48922.2021.01375.

[3]. A. Al-qerem, Y. S. Alsalman and K. Mansour, "Image Generation Using Different Models Of
Generative Adversarial Network," 2019 International Arab Conference on Information Technology (ACIT), Al Ain, United Arab Emirates, 2019, pp. 241-245, doi: 10.1109/ACIT47987.2019.8991120.

[4] https://keras.io/examples/vision/cutmix/

[5]https://www.tensorflow.org/tutorials/generative/pix2pix

[6]https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/