Generative Neural Network Based Image Compression

Aryan Gupta 2020101091 | Pratham Gupta 2020101080 | Anmoldeep Kaur Dhillon 2020101085 | Siddh Jain 2020101082

1 Introduction & Motivation

Traditional lossy image compression techniques such as JPEG and WebP are not data-specific, i.e. they are not designed specifically to handle individual datasets, in which the images are semantically related to each other. Hence these techniques do not make use of the semantic relations among the images in a specific dataset, and do not achieve the best possible compression rates. This has led to a growth in the research towards deep neural network based compression architectures. These models tend to achieve orders-of-magnitude better compression rates while still maintaining higher accuracy and fidelity in their reconstructions.

2 Literature Review

2.1 Variational Autoencoders

Given an input dataset x, characterized by an unknown probability distribution P(x), the objective is to model or approximate the data's true distribution P using a parametrized distribution p having parameters k . Let z be a random vector jointly-distributed with x. Conceptually z will represent a latent encoding of x and hence the model trains to minimize the error between x and D(z), where D(z) is the image decoded from z. [Refer: 1]

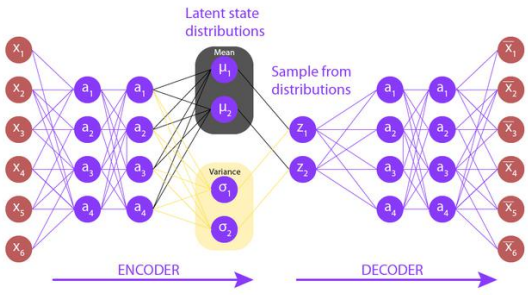

Figure 1: Variational Autoencoder

2.2 Generative Adversarial Networks:

Generator Adversarial models (GANs) are learning models which consist of a combination of the generator model and a Discriminator model working in tandem to achieve better learning. The generator model takes in random noise as input as attempts to faithfully recreate the image it is training again, while the discriminator is tasked with identifying which images were created by the generator and which were the original images. Hence, these models have a capacity to learn together and affect each others' learned parameters. [Refer: 2]

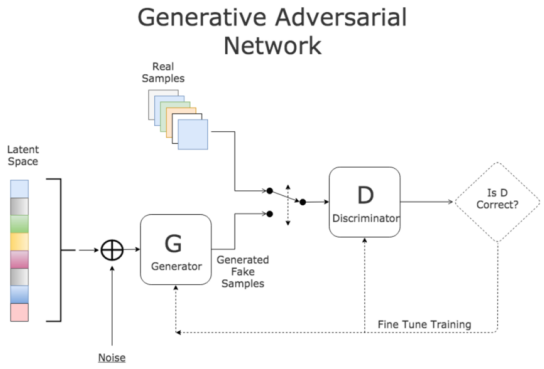

Figure 2: Generative Adversarial Network

2.3 VAE Encoder - GAN Decoder:

Santurkar et al. [2017] combine Generative Adversarial Networks (GANs) with Variational Autoencoders (VAEs) in their compression scheme, where the decoder of the compressor is the generator part of the trained GAN, which is later combined with the encoder of the VAE. By constructing the compressor with an encoder obtained from a VAE and a decoder from a GAN, authors aim to make use of the strengths of both models. Specifically, GANs are known to produce high quality, perceptual outputs that are different from the training data, whereas VAEs output images that have high fidelity to their originals, though their reconstructions are not necessarily as visually appealing to humans due to the pixel-wise loss they use for training. In a compression scheme, we need our reconstructions to be both high-quality perceptual images and to be true to their originals. This is why the literature tries to combine these two models. [Refer: 3]

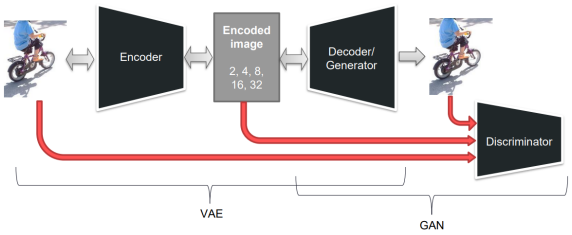

Figure 3: VAE Encoder - GAN Decoder

3 Our Contribution

In this paper, we are aiming to design a better extreme compression scheme with two main objectives:

- the scheme must achieve higher compression rates than other standard lossy image compression techniques,
- the reconstructed images must still be of high perceptual quality and true to their originals.

To assess how well our scheme is doing based on these objectives, we have to use suitable metrics. For image quality, the metrics that are relevant to our work and that are most used in the literature include the **mean squared error (MSE), peak signal-to-noise ratio (PSNR), and structural similarity index (SSIM)**. While MSE seems like the easiest metric to use, it is actually not as indicative of perceptual or textural quality as the other metrics (Wang and Bovik [2009]). On the other hand, SSIM, proposed in Wang et al. [2004], has been shown to be more indicative of

perceptual quality and thus have gained traction as the metric to use in applications where texture and perception are important.

To measure how much compression our scheme is achieving, we use the bits per pixel (BPP) metric, which conveys the magnitude of the compression scheme by indicating the (average) number of bits needed to represent one pixel. To put the numbers in context, a non-compressed image that represents each color channel of the pixel using one byte, has a BPP of 24. However, for fairness, before reporting the BPP, we losslessly c ompress the images from any scheme similar to what we do with the latent vectors in our GAN Reversal approach. Therefore, the BPP of the uncompressed PNG images will be less than 24 in our reported results on the test set. In other words,

$BPP = \frac{S_{compressed}}{512 \times 512}$,

where $S_{compressed}$ is the size of compressed file, itself losslessly compressed. Moreover, we report the compression ratio (CR), which is defined as

$CR = \frac{BPP_{uncompressed}}{BPP_{compressed}}$

4 Methodology

New Model - Fitting GAN generated images to Image at hand in latent space, omitting the VAE encoder altogether:

4.1 Experiments and Dataset

The GAN architecture is based on the CelebA faces dataset. And for the test sets, 10 images were used from the dataset. These images are face-centered and resized to the size of 512 X 512 pixels.

The experiments are run with Pytorch using stochastic gradient descent and learning rate as 1. The latent space of dimension is 512 for this GAN which gives us output image with sizes of 512 X 512. The elements of the latent vector are float32. We are able to recover the latent vector with high perceptual quality.

For the images we can see that the output of the GAN Reversal is almost perfect. This is due to the SSIM loss function being used. These results were not observed using the MSE loss and pixel-wise loss. The SSIM loss function is empirically better for this purpose.

4.2 Training Phase:

1. The dataset is split into the training and validation sub-datasets. The network tries to minimize an adversarial loss function. Generator tries to create images that cannot be differentiated from true images. Discriminator tries to correctly classify images as real or generated.

2. Discriminator is constructed just for the training purposes and is discarded after training.

[Refer: 4]

4.3 Compression Phase:

1. The specific test image is then taken (chosen randomly from the created validation dataset). A new image is also generated from random noise using the latent vectors (learned parameters) which were saved in the previous step

2. A second generator model takes this generated image and minimise the difference between the generated image and the original image so as to reproduce the original image as much as possible

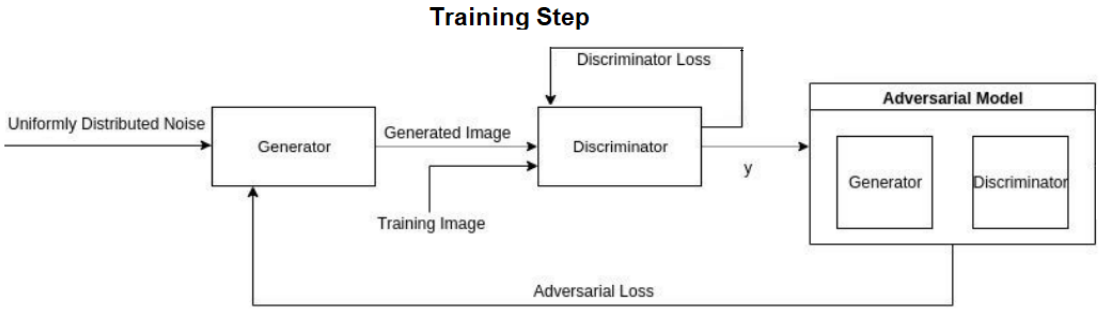

Figure 4: The architecture of Training Phase Of the Model

3. Since the generated image was generated through the latent vectors, it reduces the number of epochs required, minimises the difference between the two images and also successfully compresses the image.
4. The output image is then the compressed version of the provided input image with low loss (difference between two images)
5. Perceptual similarity metrics (Structural Similarity Index (SSIM)) used for tuning.

[Refer: 5]

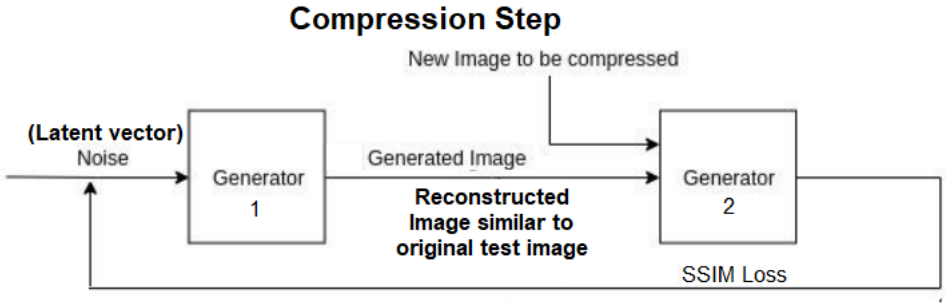

Figure 5: The architecture of Compression Phase Of the Model

4.4 Image Quality Metrics

Image quality metrics are useful for us to measure how well the architecture has performed and to define the loss.
MSE(l2) calculates the addition of squared differences between pixel values of two images. While l1 is the addition of differences between pixel values.

$$\ell_2(z^{'}) = \|f(z) - f(z^{'})\|_2^2 \quad \ell_1(z^{'}) = \|f(z) - f(z^{'})\|_1$$

These metrics are not always a good metric to access image compression quality since they do not take into account the range of variations in pixel values in an image and high, low-frequency components in an image. These factors, however, affect human perception towards quality.

4.4.1 SSIM: Structural Similarity Index Measure

SSIM calculates quality degradation in an image due to image processing tasks such as compression. SSIM is considered a better metric to access degradation of images because it takes into

account visible structures of an image. SSIM is calculated using a combination of variance and covariance terms between two images.
So the perceptual similarity metric, SSIM used for training.
The Metric Equation for Structural Similarity Index (SSIM) used to compare two images is given by:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

A suggested Loss Function to be minimised is given by:

$$\frac{1}{N}\sum_{x,y} 1 - SSIM(x;y)$$

5 Test-cases

Figure 6: Test Sample 1 - Compression using different methods

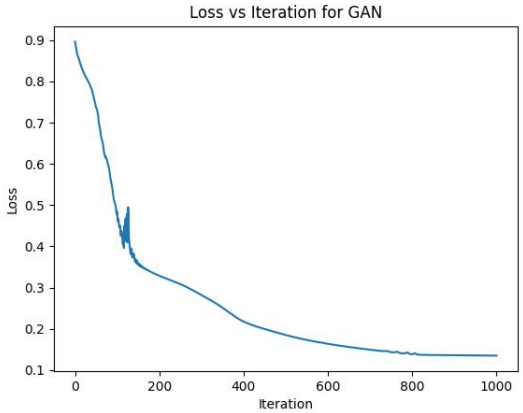

Figure 7: Test Sample 1 - SSIM Loss during Compression

6 Result and Discussion

Images when compressed using standard lossy techniques lose fidelity to the original image. The main aim of the paper was to achieve better perceptual fidelity to the original image using a neural network based approach.

Scheme	BPP	CR	PSNR	MSE	SSIM
PNG	9.860962	1	100	0	1
K-Means	0.52951	18.62279	26.05603	161.2417	0.805329
JPEG 10%	0.232788	42.36025	30.80387	54.03722	0.853143
JPEG 1%	0.15976	61.72378	23.50468	290.1435	0.689831
GAN	0.085297	115.6079	25.22288	195.3406	0.943263

Figure 8: Test Sample 2 - Metrics obtained from the baselines and "GAN Reversal"

Figure 9: Test Sample 2 - Compression using different methods

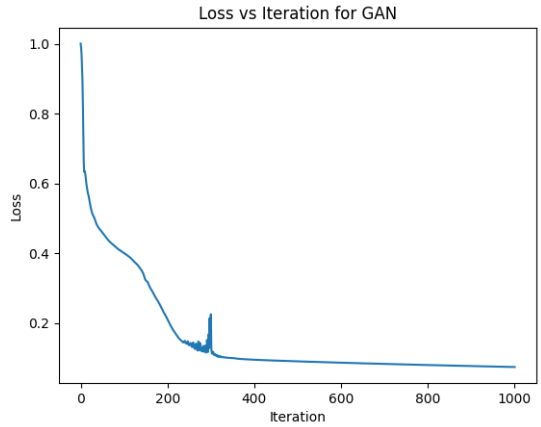

Figure 10: Test Sample 2 - SSIM Loss during Compression

Scheme	BPP	CR	PSNR	MSE	SSIM
PNG	9.860962	1	100	0	1
K-Means	0.52951	18.62279	26.05603	161.2417	0.805329
JPEG 10%	0.232788	42.36025	30.80387	54.03722	0.853143
JPEG 1%	0.15976	61.72378	23.50468	290.1435	0.689831
GAN	0.085297	115.6079	25.22288	195.3406	0.943263

Figure 11: Test Sample 2 - Metrics obtained from the baselines and "GAN Reversal"

We are successfully able to achieve this. We used two values for the quality parameter of the JPEG scheme: 10% and 1%. Although for the purposes of this paper where we are aiming for extreme compression, our main objective is beating only the JPEG 1% baseline in the SSIM metric

Figure 12: Test Sample 3 - Compression using different methods

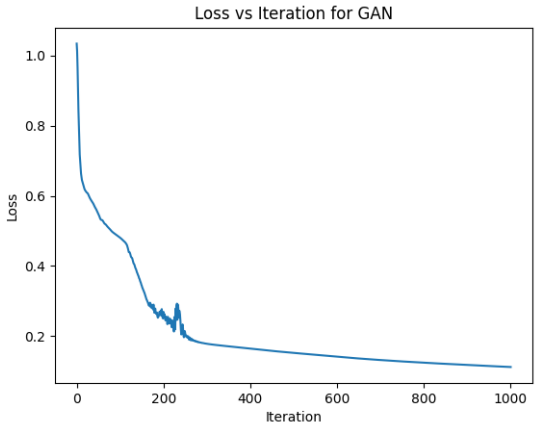

Figure 13: Test Sample 3 - SSIM Loss during Compression

Scheme	BPP	CR	PSNR	MSE	SSIM
PNG	10.46399	1	100	0	1
K-Means	0.681488	15.35462	25.01707	204.8203	0.785075
JPEG 10%	0.234222	44.67544	30.68501	55.53664	0.852781
JPEG 1%	0.161041	64.97707	22.55167	361.3379	0.700872
GAN	0.085297	122.6776	27.27924	121.6623	0.910915

Figure 14: Test Sample 3 - Metrics obtained from the baselines and "GAN Reversal"

which corresponds to better perceptual quality, the SSIM metric calculated using our model, beats all discussed lossy compression techniques(JPEG-10%, JPEG-1%, K-means), while having a better compression ratio (CR). [Tables: 8, 11, 14]

7 Conclusion

We obtain high fidelity of compressed images when the GAN is trained on a dataset of similar images, giving us a low error. When a noise vector is trained in the latent space using the Reverse-GAN scheme on this model, we obtain a low dimension latent file that produces high fidelity reconstructed image preserving substantial compression ratio. In order to learn the latent vectors and use them subsequently to compress a wide range of images, the model can also be expanded to train on other data sets. The outcome also includes a comparison of different loss functions employed by compression techniques.

8 Division of work

Literature Reading	
Generative neural Networks	Aryan,Anmol
Prior work in Image Compression	Anmol,Siddh
AutoEncoders, Decoders	Siddh,Pratham
Various Loss Functions	Pratham
Comparision of different metrics	Siddh
SSIM metric	Pratham,Aryan
Model	
Data Loading	Pratham,Siddh
Training on different databases (CelebA, CIFAR10)	Aryan,Anmol
SSIM loss neural network	Aryan,Anmol
Latent Vector Extraction	Siddh,Aryan
Compressor Function using SSIM loss	Aryan,Pratham
Utility Functions	Pratham,Anmol
Analysis	
Compressing using baseline methods	Pratham,Siddh
Evaluating five metrics	Pratham,Siddh
Performance Analysis	Siddh, Aryan
Code Piplining	All
Report Making	All

References

[1] Generative Neural Network Based Image Compression, E. Meltem Tolu- nay, Ahmad Ghalayini,
http://cs229.stanford.edu/proj2018/report/44.pdf

[2] New Losses for Generative Adversarial Learning, Victor Berger and Mich'ele Sebag, 2018,
https://arxiv.org/abs/1807.01290

[3] Understanding SSIM ,Jim Nilsson, Tomas Akenine-Möller, 2020
https://arxiv.org/abs/2006.13846