Generative Neural Network Based Image Compression

E. Meltem Tolunay
Department of Electrical Engineering
Stanford University
Stanford, CA 94305
meltem.tolunay@stanford.edu

Ahmad Ghalayini
Department of Electrical Engineering
Stanford University
Stanford, CA 94305
ahmad2@stanford.edu

Abstract

Traditional off-the-shelf lossy image compression techniques such as JPEG and WebP are not designed specifically for the data being compressed, and therefore do not achieve the best possible compression rates for images. In this paper, we construct a deep neural network based compression architecture using a generative model pretrained with the CelebA faces dataset, which consists of semantically related images. Our architecture compresses related images by reversing the generator of a GAN, and omits the encoder altogether. We report orders-of-magnitude improvements in the compression rate, compared to standard methods such as the high-quality JPEG, and are able to achieve comparable compression magnitudes to the 1%-quality JPEG, while maintaining a much higher fidelity to the original image and being able to create much more perceptual reconstructions. Finally, we evaluate our reconstructions with MSE, PSNR, and SSIM measures, compare them with JPEG of different qualities and K-means compression, and report compression magnitudes in bits per pixel (BBP) and the compression ratio (CR).

1 Introduction and related work

Standard 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. 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. Another end-to-end Convolutional Neural Network for image compression is proposed by Jiang et al. [2017]. Additionally, Theis et al. [2017] introduce Compressive Autoencoders, where they deal with the problem of the non-differentiable nature of the quantization component of compression. A quite recent paper by Agustsson et al. [2018] shows the power of incorporating GANs into these compression schemes.

Another challenge in constructing a generative deep neural compressor rises from the fact that GANs lack the encoder function. The generator network of a GAN can map from the smaller dimensional latent space to the larger dimensional image space, but not the other way around. In compression language this means that a GAN can give us a decoder but not an encoder. Addressing this issue is the work of Lipton and Tripathi [2017], where the authors train a randomly initialized vector to recover the latent space representation of an image.

Preprint. Work in progress.

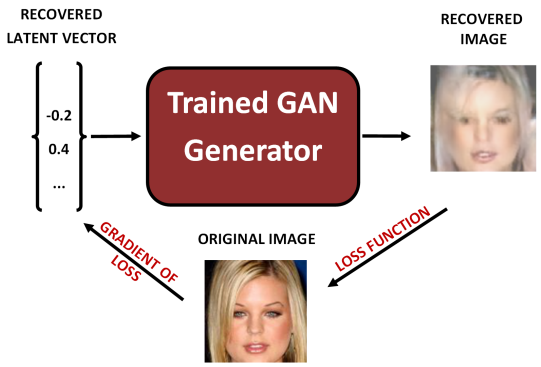

Figure 1: A simplified graphical explanation of the GAN Reversal process

2 Our contributions

Our main contribution in this paper is the introduction of this novel method of latent space vector recovery into the compression literature. Accordingly, we construct a compressor using solely a pretrained GAN generator, omitting the encoder altogether. We refer to this method as "GAN Reversal" throughout the paper. Compression is done via training a vector in the latent space, which is further compressed with bzip2, a standard lossless compression scheme. Decompression of images is simply done with a forward propagation of the latent vector through the GAN generator. To the best of our knowledge, we are not familiar with any other literature that uses this GAN Reversal scheme for image compression.

Furthermore, the mentioned paper of Lipton and Tripathi [2017] experiments with the standard ℓ_2-loss function for latent vector recovery. Our experiments indicate some limitations with this loss function. Thus, our other contribution in this paper is to combine this GAN Reversal method with another more perceptual loss function based on Structural Similarity Index (SSIM), which significantly improves some of our results that we discuss further in the paper.

3 Models and methods

A GAN consists of two networks called the generator and the discriminator. During training, the network tries to minimize an adversarial loss function. To achieve this the generator tries to create images that cannot be differentiated from true images while the discriminator tries to correctly classify images as real or generated. The discriminator is constructed just for the training purposes and is discarded after training. As stated in the introduction, the remaining generator network maps from a low dimensional latent space to a higher dimensional image space. For the formulas below, we will refer to the latent space as the z-space, and refer to the image space as the X-space. It is not an inherent feature of a GAN network to perform a mapping from $X \rightarrow z$. However, this mapping is exactly the necessary encoding step of compression. In this compression architecture, we encode images to the latent space based on the work of Lipton and Tripathi [2017]. Accordingly, a randomly initialized vector from the z-space is trained with stochastic gradient descent, whose reconstruction is aimed to be as close to the original image as possible with respect to an appropriate loss function. The mentioned paper uses a pixel-wise ℓ_2-loss function as a similarity measure of the reconstruction to the original image. Additionaly, we experiment with two other loss functions, the pixel-wise ℓ_1-loss and the SSIM metric.

The specific model construction is as follows: First, we either train a GAN or acquire the pretrained generator of a GAN, that is capable of generating images of a specific domain, such as human faces. The weights of this generator network, which exactly corresponds to the decoder of our compressor, are kept frozen. The GANs that we use for image generation are from a specific category called DCGANs introduced by Radford et al. [2015], where the sampling in the latent space is performed over a uniform distribution from the interval [-1, 1]. With this prior information, we first randomly generate a vector $z^{'}$ in the latent space, where the elements of the vector are sampled at random from a uniform distribution over the mentioned interval. Now let the mapping of the latent vector be

Figure 2: a. The original uncompressed image (26.966 bytes) which is a GAN generator output. b. JPEG compression (left: reduced to 10% quality: 1671 bytes, right: reduced to 1% quality: 661 bytes). c. Our method (standard quantization: 478 bytes, extreme quantization: 240 bytes)

$f(z^{'})$ in the X-space, and define the unknown latent vector as z. With this terminology, our original image to be compressed is simply $f(z)$. The goal of the compression is to find the vector $z^{'}$ such that $f(z^{'})$ is as close as possible to $f(z)$. This closeness measure can be defined using the following loss functions, that can further be combined with each other using empirical weighted sums as well. The pixel-wise losses that we use are the ℓ_2 (Mean Squared Error - MSE) and ℓ_1 loss functions. Hence, the possible pixel-wise loss function to be minimized are:

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

Because these mentioned loss functions are pixel-wise distance metrics, they have limitations in terms of outputting perceptual images and recovering the edges in the original images. This motivates us to use perceptual similarity metrics for our training. One of such metrics is the well-known Structural Similarity Index (SSIM). For two aligned windows x and y from different images, this metric is defined as (Wang et al. [2004]):

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

Note that this function takes the neighboring pixels into account (Zhao et al. [2017]), and it is a more perceptual metric than pixel-wise metrics. Another remark is that SSIM value increases up to 1 as the images become more and more similar, so the corresponding loss function to be minimized is:

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

We minimize these loss functions with respect to the latent vector $z^{'}$ via stochastic gradient descent. Note that we actually know that the unknown latent vector was sampled from U[-1, 1]. Thus after each iteration, we can clip the vector to stay in this range. Another important remark is that this GAN reversal training is non-convex, and we cannot guarantee to recover the same latent vector after each training. Multiple latent vectors can indeed map to the same image, but for our compression purposes it does not matter which latent vector we recover, as long as its corresponding image is close to the original. Figure 1 shows a simplified graphical explanation of GAN Reversal.

4 Experiments and evaluation

4.1 Experiments and dataset

In our initial experiments, we use a GAN architecture implemented according to the work of Karras et al. [2017], pretrained on CelebA faces dataset (Liu et al. [2015]).

For the test set, and for an unbiased evaluation of the models and baselines used, we used a test set of 10 images from the CelebA dataset, face-centered and resized to be of size 128×128 pixels. The images in the test set are not outputs of the GAN generator (unlike in Figure 2, where the image was actually an output of the GAN generator for some latent vector). Accordingly, we run our initial experiments with "seen" images that have been created by the GAN itself in the first place. This is a natural starting point considering that we know that the GAN is capable of outputting these images. Later we move on to "unseen" images that are taken from the CelebA test set as we mentioned earlier.

Figure 3: a. An uncompressed unseen image from our test set (29.171 bytes). b. K-means ($K = 4$) compressed image c. JPEG compression (left: reduced to 10% quality, 1.030 bytes; right: reduced to 1% quality, 457 bytes). c. Our GAN Reversal scheme (extreme quantization: 450 bytes)

All experiments are run with TensorFlow, using stochastic gradient descent with an empirical learning rate between 1 and 10. The particular GAN model we use has a latent space dimension of 512 and outputs images of size $128 \times 128 \times 3$. All elements of the latent vectors are `float32` numbers. After training and recovering the latent vector corresponding to the original image, we perform additional quantization. Our experiments show that the latent vectors can be rounded up to 1 significant figure after the decimal point without any significant perceptual quality in the decompressed image for both seen and unseen images. For seen images, a vocabulary size of 21 is in practice sufficient to store the image in the latent space, where $z_i^{'} \in \{-1.0, -0.9, -0.8, ..., 0.0, ..., 0.9, 1.0\}$. This is because for seen images the training will almost always succesfully recover the latent vector within an interval of [-1,1]. Furthermore, if we round the latent vectors just to the set of $\{-1, 0, 1\}$, we still construct images that have very high perceptual qualities. However, high fidelity is not always guaranteed for this extreme case of quantization. This small vocabulary size in the latent space also allows us to perform efficient lossless compression on top of quantization. For this, we use an off-the-shelf `bzip2` algorithm. For unseen images, we found that the training outputs latent vectors whose elements are from a larger interval (approximately [-15,15]). Clipping during training made the fidelity of reconstructions worse, so we performed quantization in this larger interval for unseen images, and then ran the lossless `bzip2` compression.

Figure 2 illustrates results from our method for seen images. For such cases where we know the image is an output of the GAN generator, finding a latent vector that outputs that image through GAN Reversal is almost perfect, as can be seen in Figure 2, even using just MSE loss. However, the challenge lies in the images that are not exact outputs of the GAN generator, where the MSE loss is no longer sufficient, and we use the SSIM loss for the unseen images. We added a weighted sum of one of the pixel-wise losses during training, but found empirically that the SSIM loss alone works best among all for our purposes. Results for unseen images are depicted in Figure 3.

4.2 Metrics used and baseline models

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, and 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 compress 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

$Scheme$	BPP	CR	$PSNR$	MSE	$SSIM$
PNG (Uncompressed)	13.63	1	∞	0	1
K-means ($K = 4$)	1.486	9.172	23.012	342.46	0.7145
JPEG (10%)	0.4848	28.11	26.161	161.87	0.7711
JPEG (1%)	0.2176	62.64	20.465	589.15	0.5280
GAN Reversal (Our approach)	0.2152	63.34	21.192	540.06	**0.7073**

Table 1: Summary of the metrics obtained from the baselines and our approach "GAN Reversal" on our test set

images will be less than 24 in our reported results on the test set. In other words, $BPP = \frac{S_{compressed}}{128 \times 128}$, where $S_{compressed}$ is the size of compressed file, itself losslessly compressed. Moreover, we report the compression ratio (CR), which is defined as $CR = \frac{BPP_{uncompressed}}{BPP_{compressed}}$.

One of the baselines that we investigated first was compression using K-means clustering. The BPP of the K-means algorithm is: $BPP_{kmeans} \approx log_2(K)$. However, since we are also losslessly compressing the images from K-means, $BPP_{kmeans} < log_2(K)$. For this paper, we only try $K = 2$ because our aim is extreme compression.

The other baseline we investigated was JPEG (optimized), which is a popular lossy image compression scheme. We used two values for the *quality* parameter of the JPEG schem: 10% and 1%. Note that for the purposes of this paper where we are aiming for extreme compression, our main objective is beating the JPEG 1% baseline in the SSIM metric which corresponds to better perceptual quality, while still having a comparable compression ratio (CR).

4.3 Results and discussion

Table 1 summarizes the performance of the baselines as well as our approach "GAN Reversal" on the proposed metrics. Figure 3 compares the outputs of the different compression schemes on a sample image from our test set. The results show that our approach has a compression ratio (CR) comparable to that of the extreme JPEG (1%) scheme (and even delivers a smaller size in most cases), while clearly outperforming it in the SSIM metric. For seen images, we can achieve sizes approximately as small as 250 bytes. This number is slightly bigger for unseen images, with an approximate size of 450 bytes after extreme quantization. Note that it's harder to achieve higher fidelity in the reconstructions for unseen images, compared to seen ones, where the simple MSE loss almost always guarantees perfect recovery of the latent vector. We had to use a more sophisticated SSIM loss for the unseen images, which correlates much better with human perception. We believe that this new compression scheme is very promising for the future, since it utilizes the semantic relations of a group of images using a pretrained GAN, thus eliminating the redundant side information that can be stored elsewhere to allow for extreme compression.

5 Future work

The main future goal should be to improve the loss function used for latent vector training even further. This will enable us to build compressors that achieve more perceptual reconstructions with higher fidelity. The additional challenge will be to build an automated process for the compressor. Since the compression procedure utilizes a gradient descent based training scheme, there are significant parts of compression that rely on human observation; such as the hyperparameter tuning, running other training sessions with differently initialized random vectors for improvement, and picking the best perceptual output among all reconstructions. For a practical compressor, all these processes must be automated.

Acknowledgments

We would like to thank Kedar Tatwawadi and Shubham Chandak for their very helpful discussions and comments.

References

Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative adversarial networks for extreme learned image compression. *CoRR*, abs/1804.02958, 2018. URL http://arxiv.org/abs/1804.02958.

Feng Jiang, Wen Tao, Shaohui Liu, Jie Ren, Xun Guo, and Debin Zhao. An end-to-end compression framework based on convolutional neural networks. *CoRR*, abs/1708.00838, 2017. URL http://arxiv.org/abs/1708.00838.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *CoRR*, abs/1710.10196, 2017. URL http://arxiv.org/abs/1710.10196.

Zachary C. Lipton and Subarna Tripathi. Precise recovery of latent vectors from generative adversarial networks. *CoRR*, abs/1702.04782, 2017. URL http://arxiv.org/abs/1702.04782.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015. URL http://arxiv.org/abs/1511.06434.

Shibani Santurkar, David M. Budden, and Nir Shavit. Generative compression. *CoRR*, abs/1703.01467, 2017. URL http://arxiv.org/abs/1703.01467.

Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. *CoRR*, abs/1703.00395, 2017. URL http://arxiv.org/abs/1708.00838.

Zhou Wang and Alan C Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE signal processing magazine*, 26(1):98–117, 2009.

Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004. ISSN 1057-7149. doi: 10.1109/TIP.2003.819861.

H. Zhao, O. Gallo, I. Frosio, and J. Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging*, 3(1):47–57, March 2017. ISSN 2333-9403. doi: 10.1109/TCI.2016.2644865.