High-Fidelity Generative Image Compression

Fabian Mentzer, George Toderici, Michael Tschannen, Eirikur Agustsson

Google Research

Gio Paik
giopaik@naver.com

Image Compression

- There are two main categories in Image Compression.
  - Lossy Compression
  - Lossless Compression

	Lossy Compression	Lossless Compression
File Size	Relatively Small	Relatively Big
Data Loss	Lose some detail.	Keep all detail.
Formats	JPEG, WebP, BPG	PNG, GIF, PCX

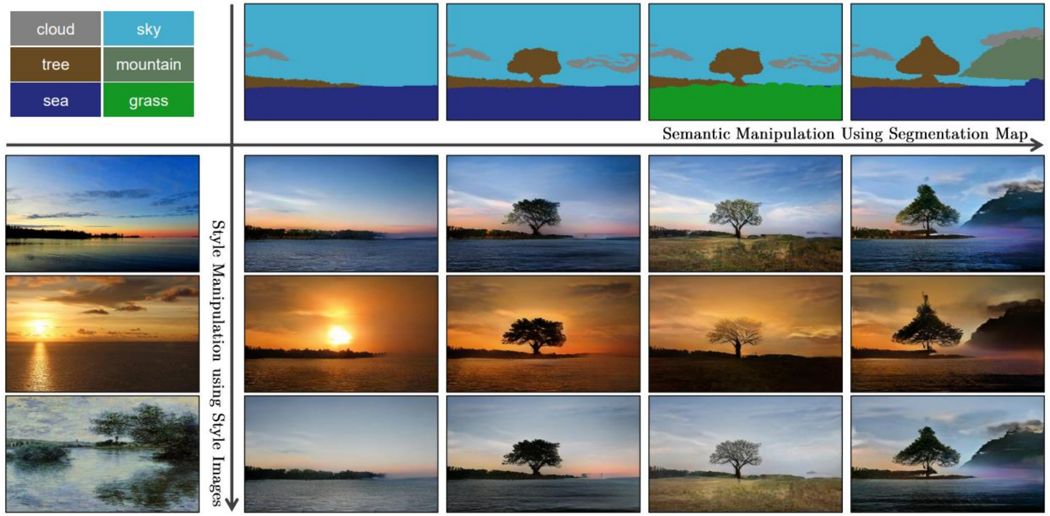
JPEG: Most frequently used lossy comp.

- JEPG is the most frequently used lossy compression method.
- JPEG method compress(=lose) data using quantization.
- Compression occurs on every pixels.

Felis_silvestris_silvestris.jpg: Michael Gäbler derivative work: AzaToth – Felis_silvestris_silvestris.jpg

Combining GAN with Image Compression

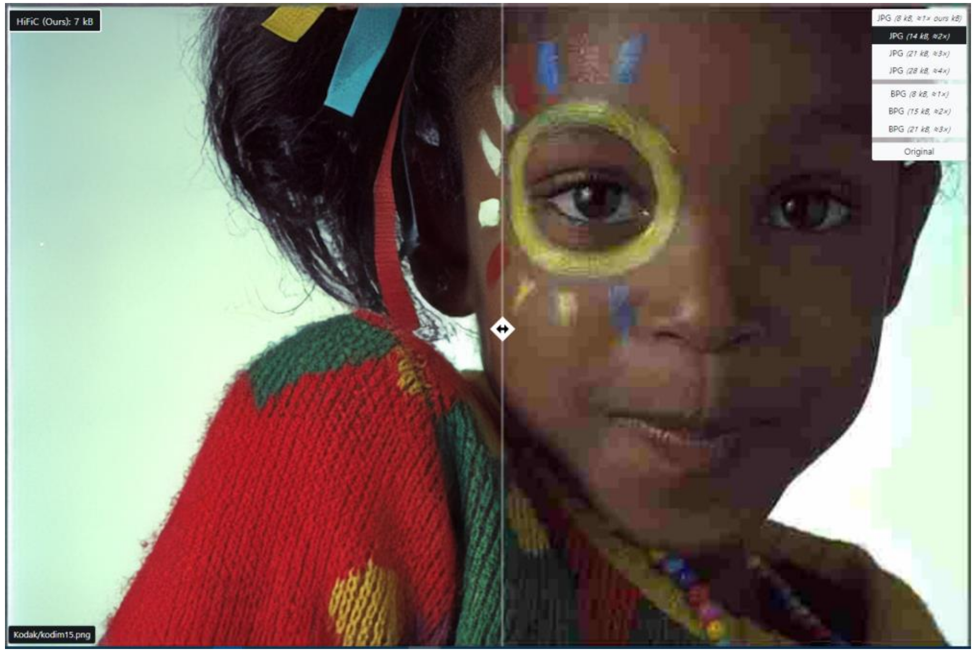
- Generative Adversarial Nets (GANs) can generate photo-realistic high resolution images.
- For an example GauGAN from Nvidia can generate realistic landscape images from simple sketch.



Taesung Park et al, Semantic Image Synthesis with Spatially-Adaptive Normalization, CVPR 2019

Combining GAN with Image Compression

- Since GAN can generate realistic images from low information, We can use GAN to compress/reconstruct image!
- High Fidelity Compression (HiFiC) : Image Compression with NN



HiFiC (ours) compared with JPG

Figure from project page

Neural Image Compression

- Lossy Compression based on Shannon's rate-distortion theory is usually modeled with an auto encoder.
- By encoding an image x, we obtain a quantized latent $y = E(x)$.
- Using decoder, we obtain the lossy reconstruction $x' = G(y)$.
- This compression incurs a distortion $d(x, x')$, e.g. $d = MSE$.
- Using probability model P of y, we can store y losslessly using bitrate $r(y) = -\log(P(y))$. (arithmetic coding)
- If we parameterize $E, G\ and\ P$ as CNNs, we can train them jointly by minimizing the rate-distortion trade-off.
$$\mathcal{L}_{EG} = \mathbb{E}_{x \sim P_x}[\lambda r(y) + d(x, x')].$$
- λ is hyper parameter to control the trade-off.

Formulation and Optimization

- We use loss functions below to train E,G,P,D. where $d_P = LPIPS$, and $d = k_M MSE + k_P d_P$, where k_M, k_P are hyper params.
- Using hyper-parameters λ, β, we obtain:

$$\mathcal{L}_{EGP} = \mathbb{E}_{x\sim p_X}\left[\lambda r(y) + d(x, x') - \beta \log\big(D(x', y)\big)\right],$$
$$\mathcal{L}_{D} = \mathbb{E}_{x\sim p_X}\left[-\log\big(1 - D(x', y)\big)\right] + \mathbb{E}_{x\sim p_X}\left[-\log\big(D(x, y)\big)\right].$$

- Now you see it's hard to making comparison because we have so many hyper-parameters that odds to each others like k_M, k_P, λ, β.
- So we use "rate target" hyper-parameter r_t to replace λ.
- If $r(y) > r_t, \lambda\ will\ be\ \lambda^{(a)}.\ and\ \lambda\ will\ be\ \lambda^{(b)}\ otherwise$.
- Setting $\lambda^{(a)} \gg \lambda^{(b)}$ allows us to learn a model with an avg bitrate close to r_t.

Model Architecture

- Architecture of encoder E, generator G, discriminator D and probability model P are shown below.
- Probability model P is based on hyper-prior model from [1].
- E, G and D are based on [2, 3], with some key differences in the D and in the normalization layers.

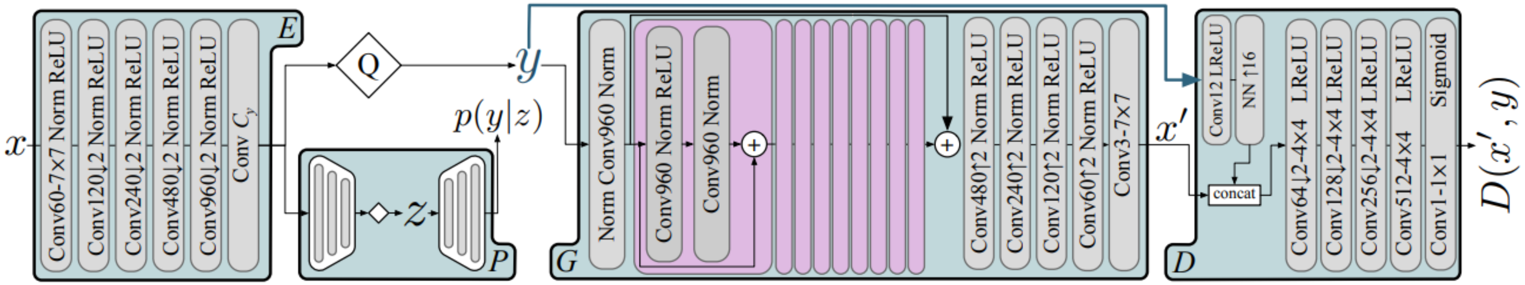


Figure 2: Our architecture. *ConvC* is a convolution with C channels, with 3×3 filters, except when denoted otherwise. $\downarrow 2$, $\uparrow 2$ indicate strided down or up convolutions. *Norm* is ChannelNorm (see text), *LReLU* the leaky ReLU [53] with α=0.2, *NN*↑16 nearest neighbor upsampling, Q quantization.

[1] Johannes Ballé et al, Variational image compression with a scale hyperprior, ICLR 2018.
[2] Ting-Chun Wang et al, High-resolution image synthesis and semantic manipulation with conditional gans, CVPR 2018.
[3] Eirikur Agustsson et al, Generative adversarial networks for extreme learned image compression, ICCV 2019.

Model Architecture

- Both [2, 3] use a multi-scale patch-discriminator, while we use a single scale.
- We replace InstanceNorm with SpectralNorm [4].
- Importantly, and in contrast to [3], we condition D on y by concatenating an upscaled version to the image, as shown in Figure.

[4] Takeru Miyato et al. Spectral normalization for generative adversarial networks. ICLR, 2018.

Results

- We compared HiFiC with other lossy compression algorithms and got the result shown below.

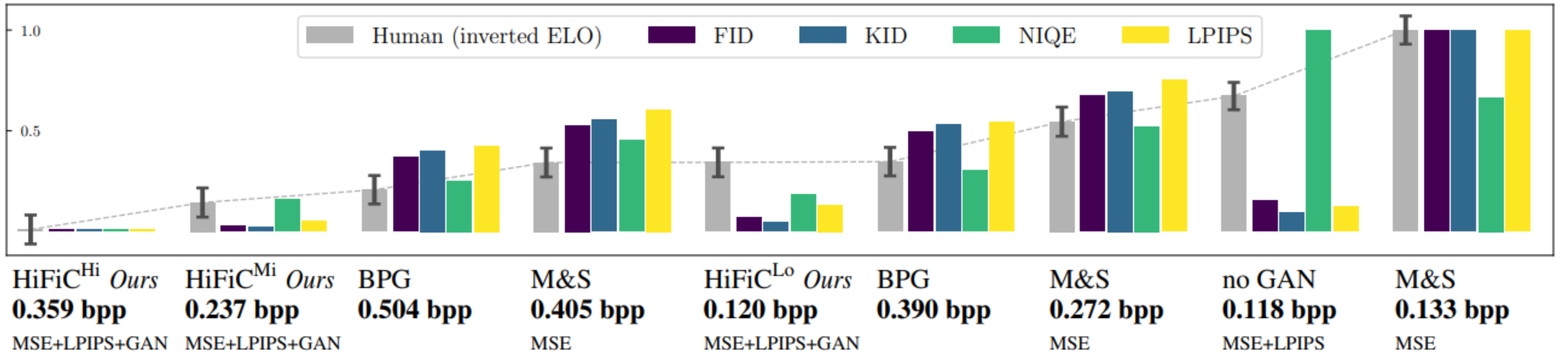


Figure 3: Normalized scores for the user study, compared to perceptual metrics. We invert human scores such that **lower is better** for all. Below each method, we show *average* bpp, and for learned methods we show the loss components. "no GAN" is our baseline, using the same architecture and distortion d as *HiFiC (Ours)*, but no GAN. "M&S" is the *Mean & Scale Hyperprior* MSE-optimized baseline. The study shows that training with a GAN yields reconstructions that outperform BPG at practical bitrates, for high-resolution images. Our model at 0.237bpp is preferred to BPG even if BPG uses $2.1\times$ the bitrate, and to MSE optimized models even if they use $1.7\times$ the bitrate.

Summary

- GANs are able to create and reconstruct realistic images.
- So we can apply GANs for Image Compression to reconstruct image with high perceptual fidelity.

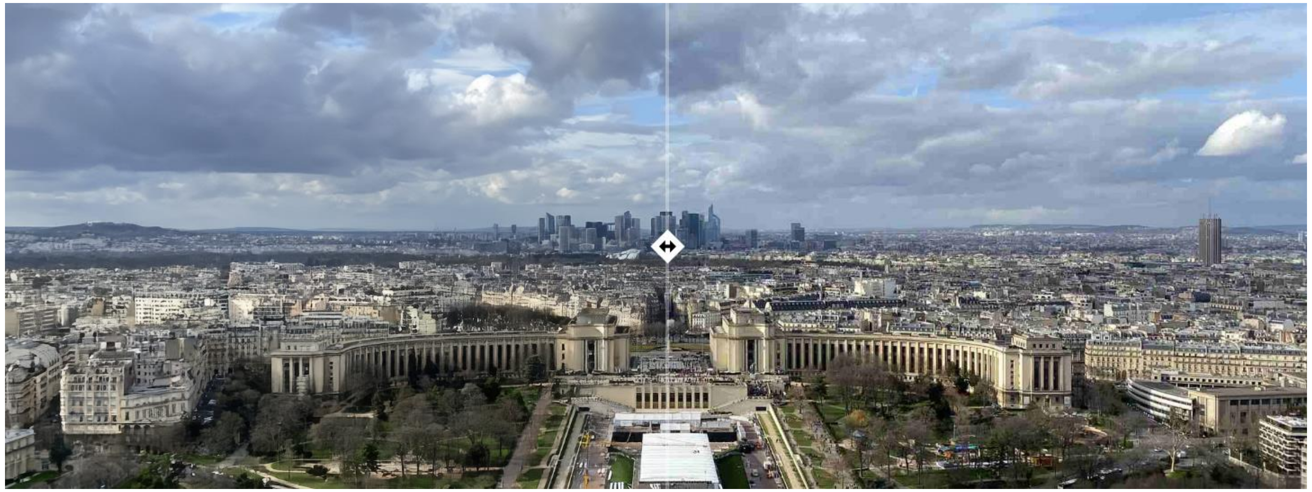


Image from project page

High-Fidelity Generative Image Compression

Official Project Page: https://hific.github.io/

Paper: https://arxiv.org/abs/2006.09965

Thank You for watching!

Gio Paik
giopaik@naver.com