Generative Adversarial Networks for Extreme Learned Image Compression

Authors: Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, Luc Van Gool

Presentation by Quan Bach

Overview

- Objective
- Main Contributions
- Key Ideas
- Methods
- Network Architecture
- Experiments
- Results
- Appendix

Objective

Problem being solved: Image compression that preserves the original image and at extremely low bitrates.

Objective: Implement a learned image compression system based on GANs that generate learned compression at extremely low bitrates.

Main Contributions

1. Provide a principled GAN framework for full resolution image compression and use it to build an extreme image compression system.
2. Thoroughly explore such a framework in the context of full-resolution image compression.
3. Set new state-of-the-art in visual quality based on a user study, with dramatic bitrate savings.

Key Ideas

- Combination of learned compression and conditional GANs $\longrightarrow$ proposed GANs framework
- Control the maximum bitrates through the upper bound of the entropy. Avoid to model the entropy explicitly as loss term. Average bits need to encode the presentation from the encoded-quantized image is measured by the entropy of that presentation.

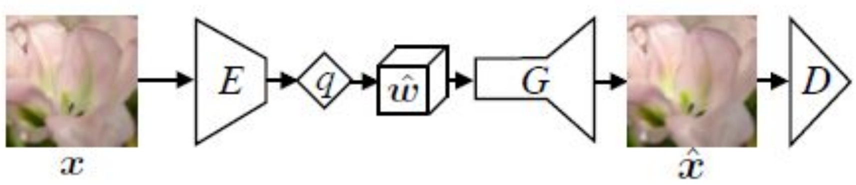
Methods

Proposed a principled GAN framework for full-resolution image compression targeting bitrates below 0.1 bits per pixel (bpp).

Two modes of operations:

- Generative Compression (GC): preserving the overall image content, while generating structure of different scale (leaves, windows, trees, etc.)
- Selective Generative Compression: completely generating parts of the image from a semantic label map, while preserving user-defined regions with a high degree of detail.

Methods: Generative Compression



E: encoder	[Appendix A.1]
q: quantizer (differentiable relaxation of q)	[Appendix A.2]
$\hat{\omega}$: quantized feature map	[Appendix A.3]
G: generator	[Appendix A.4]
D: discriminator	[Appendix A.5]

Methods: Generative Compression

Saddle-point objective for (unconditional) GC:

weights to control bitrates

$$min_{E,G}\ \mathcal{L}_{GAN} + \underbrace{\lambda \mathbb{E}[d(\boldsymbol{x}, G(\boldsymbol{z}))]}_{\text{distortion term}} + \beta \underbrace{H(\hat{\boldsymbol{\omega}})}_{\text{entropy}}$$

[Appendix B.1]

$$latent\ vector\ \boldsymbol{z} = [\hat{\boldsymbol{\omega}}, \boldsymbol{v}];\ where\ \boldsymbol{v}\ is\ noise\ drawn\ from\ fixed\ prior\ p_{\boldsymbol{v}}.$$

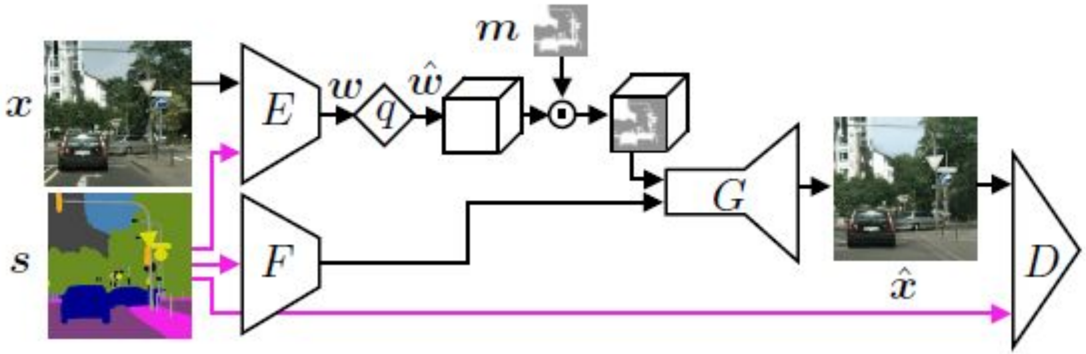
$$\mathcal{L}_{GAN} := max_D\ \mathbb{E}[f(D(\boldsymbol{x}))] + \mathbb{E}[g(D(G(\boldsymbol{z})))]$$

[Appendix B.2]

Conditional GC [GC (D+)]:

$$\mathcal{L}_{cGAN} := max_D\ \mathbb{E}[f(D(\boldsymbol{x}, \boldsymbol{s}))] + \mathbb{E}[g(D(G(\boldsymbol{z}, \boldsymbol{s}), \boldsymbol{s}))]$$

Methods: Selective Generative Compression



E: encoder

q: quantizer

$\hat{\omega}$: quantized feature map

G: generator

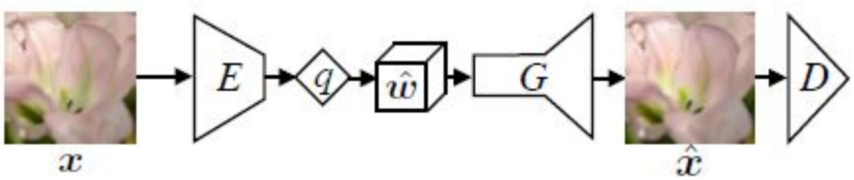
D: discriminator

s : semantic map [Appendix C.1]

F: features extractor

m: heatmap [Appendix C.2]

Network Architecture: GC

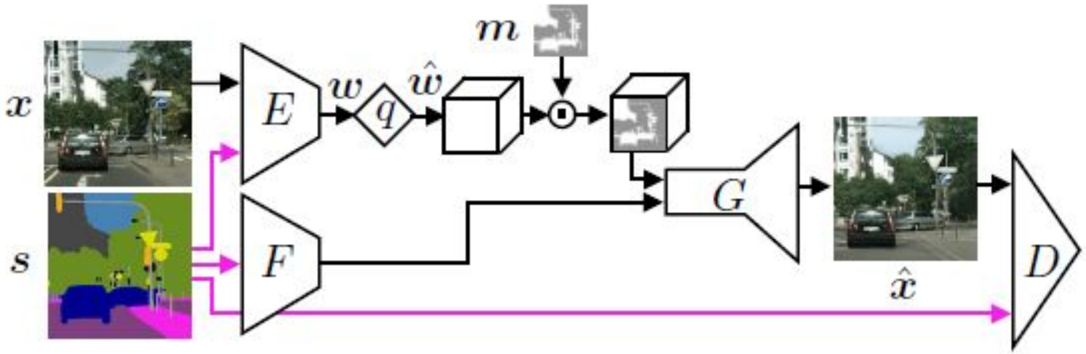


E: c7s1-60, d120, d240, d480, d960, c3s1-C, q

G: c3s1-960, R960 (x9), u480, u240, u120, u60, c7s1-3

- c7s1-k : 7x7 conv-InstanceNorm-ReLU: k filters + stride 1.
- dk : 3x3 conv-InstanceNorm-ReLU: k filters + stride 2
- Rk: RES block has two 3x3 conv layers same # of filters k
- uk: 3x3 fractional-strided-conv-InstanceNorm-ReLU: k filters + stride ½ .

Network Architecture: SC



Encoder:

Semantic map: c7s1-60, d120, d240, d480, d960.

Image encoder: c7s1-60, d120, d240, d480, c3s1-C, q, c3s1-480, d960

G: c3s1-960, R960 (x9), u480, u240, u120, u60, c7s1-3

Experiments: Hyperparameters

- β = 0
- λ = 10 (adopt MSE for the distortion term)
- L = 5
- centers C = {-2, 1, 0, 1, 2}
- Control the bitrates through the upper bound of the entropy:

$$H(\hat{\omega}) \leq dim(\hat{\omega}).log_2(L)$$

[Appendix B.1]

Objective function:

$$min_{E,G}\ \mathcal{L}_{GAN} + \lambda \mathbb{E}[d(\boldsymbol{x}, G(\boldsymbol{z}))] + \cancel{\beta H(\hat{\boldsymbol{\omega}})}$$

For GC: C = 2 $\longrightarrow$ 0.0181 bpp

$$\frac{H(\hat{\omega})}{WH} \leq \frac{WH}{16.16}.C.\frac{log_2(L)}{WH}$$

Experiments: Training details

- Optimizer: ADAM
- LR: 0.0002
- mini-bs: 1
- Cityscapes: 150K iterations
- OpenImages: 280K iterations
- Instance Normalization
- Note: for second half of OpenImages, train G with fixed batch statistics with batch norm to reduce artifacts and color-shift

Experiment: Evaluation

- GC - no semantic map: trained on 188k images from OpenImages and evaluate on Kodak + 20 random images from RAISE1K
- GC - semantic map: trained on Cityscapes using random 20 images from validation set for evaluation
- SC: train on Cityscapes

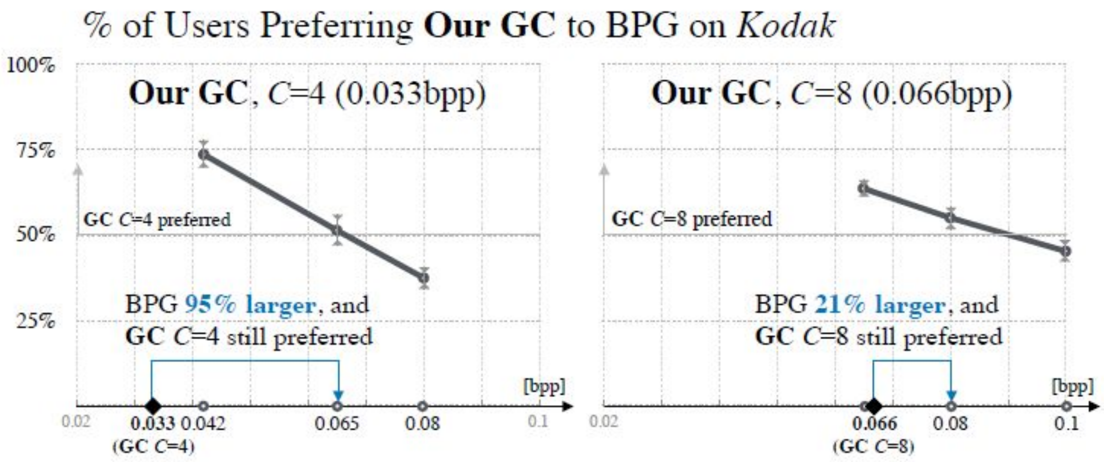
Experiment: Baselines

- HEVC-based image compression algorithm BPG (current SOTA engineered img compression codec)
- AEDC network - trained with bottleneck depth C= 4 for MS-SSIM on CItyscapes with Early stopping (originally was trained on ImageNet) [0.07bpp]

Experiment: User-study

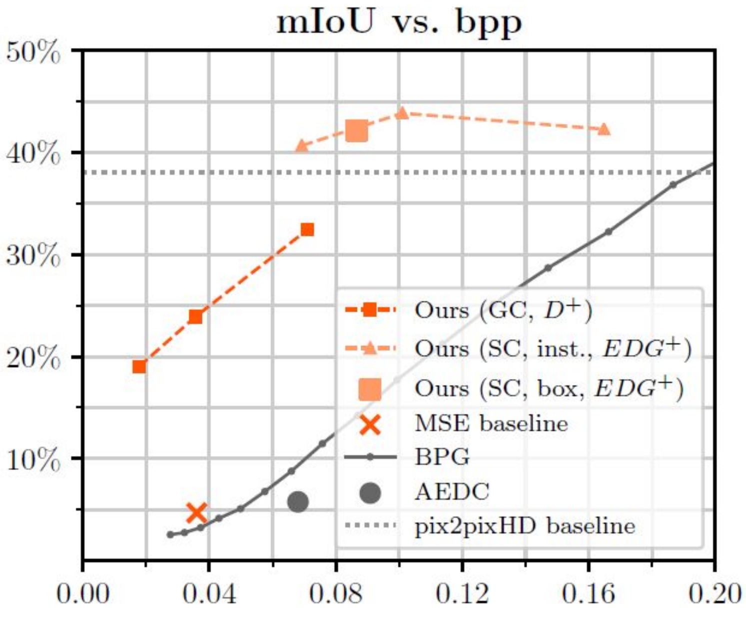
- GC: 2 models - C = 4, 8 on OpenImages
- GC [D+] : 3 models - C = 2, 4 ,8 on CityScapes
- BPG: 0.045 $\longrightarrow$ 0.12 bpp

Results



User study results evaluating our GC models on Kodak, RAISE1K and Cityscapes. The bitrate of the model is highlighted on the x-axis with a black diamond. The thick gray line shows the percentage of users preferring GC model to BPG at that bitrate (bpp). The blue arrow points from GC model to the highest-bitrate BPG operating point where more than 50% of users prefer GC, visualizing how many more bits BPG uses at that point. More at [Appendix D.1, D.2]

Results



Mean IoU as a function of bpp on the Cityscapes validation set for GC and SC networks, and for the MSE baseline. They show both SC modes: RI (inst.), RB (box). D+ annotates models where instance semantic label maps are fed to the discriminator (only during training); EDG+ indicates that semantic label maps are used both for training and deployment.The pix2pixHD baseline was trained from scratch for 50 epochs, using the same downsampled 1024 x 512px training images as for their method.

Conclusion

- A thorough study of a learned compression framework for full-resolution image compression
- User-study to evaluate the results instead of classical MS-SSIM and MSE.
- Demonstrated that constraining the application domain to street scene images leads to additional storage savings
- Explored (for SC) selectively combining fully synthesized image contents with preserved one when semantic label maps are available.
- Future works:
  - develop a mechanism for controlling spatial allocation of bits for GC

Appendix A.1 : Encoder

- They use an arithmetic encoder to encode the channels of $\hat{\omega}$ to a bit-stream, storing frequencies for each channel separately
- leads to 8.8% smaller bitrates compared to the upper bound
- For the GC, the encoder E convolutionally processes the image x and optionally the label map s, with spatial dimension W x H, into a feature map of size W/16 x H/16 x 960 (with 6 layers, of which four have 2-strided convolutions), which is then projected down to C channels (where C = 2, 4, 8 is much smaller than 960).

Appendix A.2 : Quantizer

To be able to backpropagate through the non-differentiable q, one can use a differentiable relaxation of q. Given centers C = {c1, · · · , cL} ⊂ R, we use nearest neighbor assignments to compute

$$\hat{z}_i = Q(z_i) := \arg\min_j \|z_i - c_j\|,$$

but rely on (differentiable) soft quantization

$$\tilde{z}_i = \sum_{j=1}^{L} \frac{\exp(-\sigma\|z_i - c_j\|)}{\sum_{l=1}^{L} \exp(-\sigma\|z_i - c_l\|)} c_j$$

Appendix A.3: quantized feature map

- Sampling the compressed representations: explore the representation learned by our GC models (with $C = 4$), by sampling the (discrete) latent space of $\hat{\omega}$ $\rightarrow$ “soup of image patches” which reflects the domain the models were trained on.
- Perform a simple experiment and train an improved Wasserstein GAN (WGAN-GP). By feeding our GC model with samples from the WGAN-GP generator $\rightarrow$ obtain a powerful generative model, which generates sharp 1024 x 512 px images from scratch.

$\mathcal{U}$ (Open Images) $\mathcal{U}$ (Cityscapes) WGAN-GP (Cityscapes)

Appendix A.4: Generator

- The generator G projects $\hat{\omega}$ up to 960 channels, processes these with 9 residual units at dimension W=16 H=16 960, and then mirrors E by convolutionally processing the features back to spatial dimensions W H (with transposed convolutions instead of strided ones).

Appendix A.5: Discriminator

- Discriminator computes the same f-divergence for the objective function of GANs.
- use the multi-scale architecture for the discriminator D, which measures the divergence between p_x and p_G(z) both locally and globally.
- To differentiate high-resolution real and synthesized images, the discriminator needs to have a large receptive field.
- Use 3 discriminators that have an identical network structure but operate at different image scales.
- Downsample the real and synthesized high resolution images by a factor of 2 and 4 to create an image pyramid of 3 scales. The discriminators D1, D2 and D3 are then trained to differentiate real and synthesized images at the 3 different scales, respectively

Appendix B.1: Entropy

- The average number of bits needed to encode $\hat{\omega}$ is measured by the entropy $H(\hat{\omega})$, which can be modeled with a prior [1] or a conditional probability model [2].
- $H(\hat{\omega})$ is bounded by the upper bound (from Fanon's inequality*)[3] :

$$H(\hat{\omega}) \leq dim(\hat{\omega}).log_2(L)$$

[1]: Soft-to-Hard Vector Quantization for End-to-End Learning Compressible Representations: https://arxiv.org/pdf/1704.00648.pdf

[2]: Conditional Probability Models for Deep Image Compression: https://arxiv.org/pdf/1801.04260.pdf

[3]: Thomas M Cover and Joy A Thomas. Elements of information theory. John Wiley & Sons, 2012.

*still need to double check.

Appendix B.2: GANs objective function

GANs objective function:

$$\mathcal{L}_{GAN} := max_D \ \mathbb{E}[f(D(\boldsymbol{x}))] + \mathbb{E}[g(D(G(\boldsymbol{z})))]$$

- Nowozin et al. [1] show that for suitable choices of f and g solving minG LGAN allows to minimize general f-divergences between the distribution of G(z) and p_x.
- adapt Least-Squares GAN [2] in this paper. Corresponds to Pearson χ^2 divergence:

$$f(y) = (y-1)^2 \ and \ g(y) = y^2$$

Appendix C.1: Semantic map

- Can be obtained using off-the-shelf semantic/instance segmentation networks, e.g., PSPNet and Mask R-CNN.
- SC requires a semantic/instance label map of the original image
- Constrain the fully synthesized regions to have the same semantics **s** as the original image **x**.

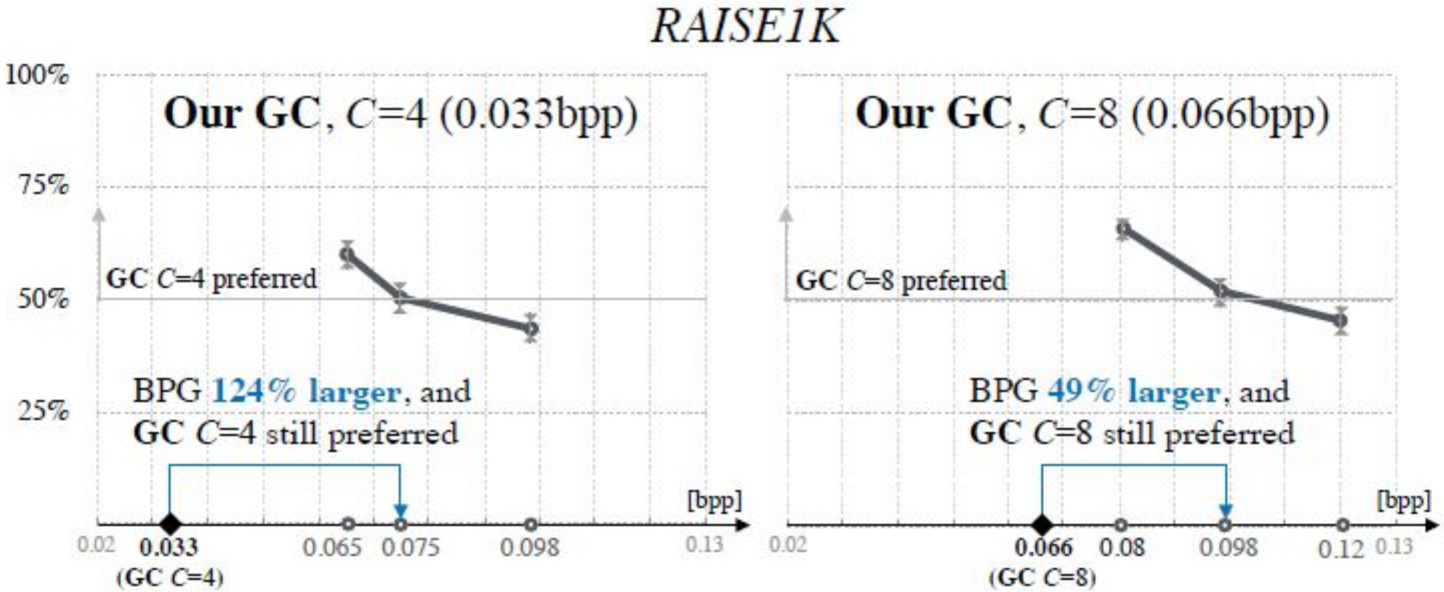
Appendix C.2: Heatmap

- Binary: 1’s to be preserved and 0’s to ignore.
- Heatmap m is also stored, only encode the entries of $\hat{\omega}$ corresponding to the preserved regions.

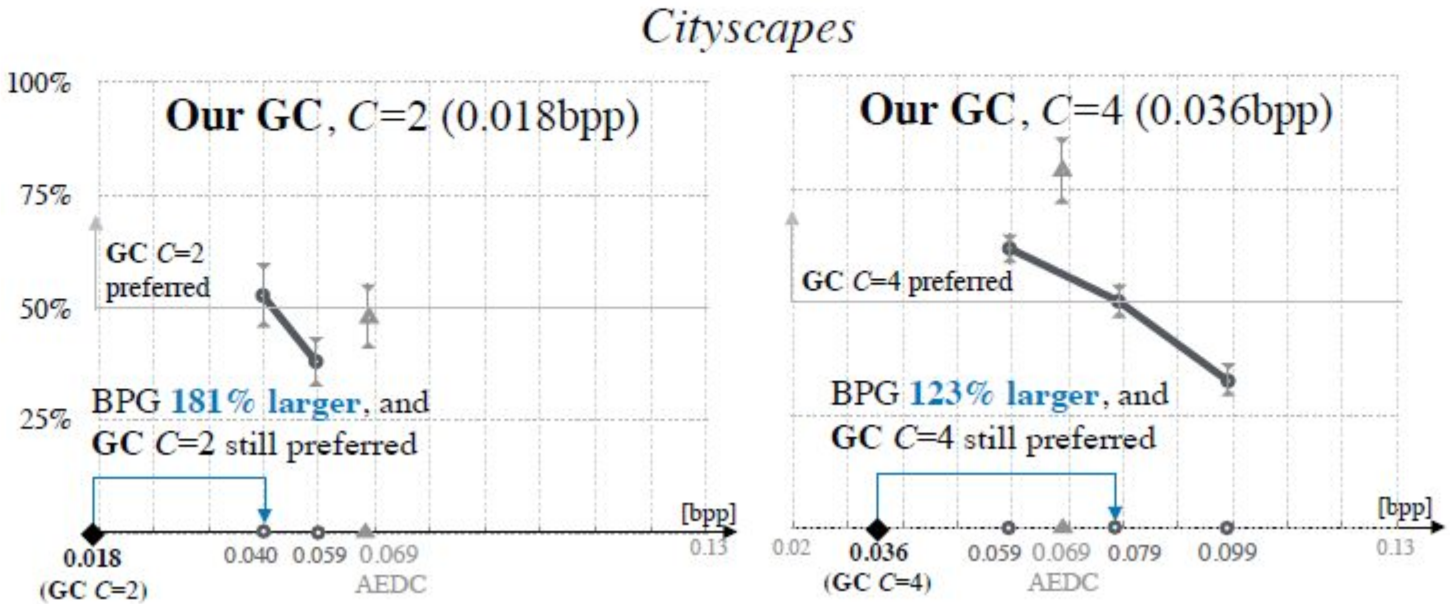
road (0.146bpp, -55%) | *car* (0.227bpp, -15%) | all synth. (0.035bpp, -89%)

people (0.219bpp, -33%) | *building* (0.199bpp, -39%) | no synth. (0.326bpp, -0%)

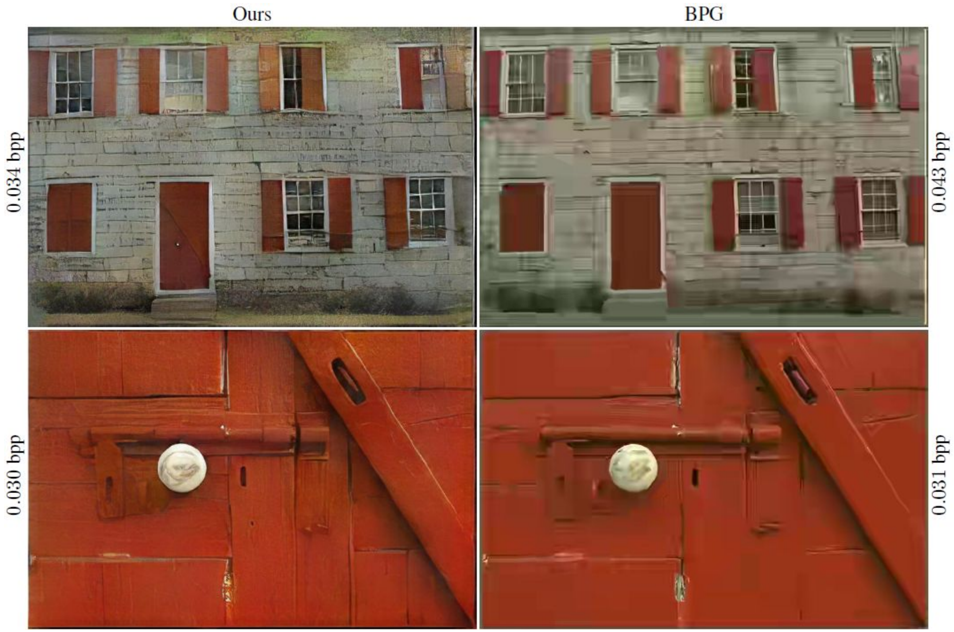
Appendix D.1: RAISE1K Results



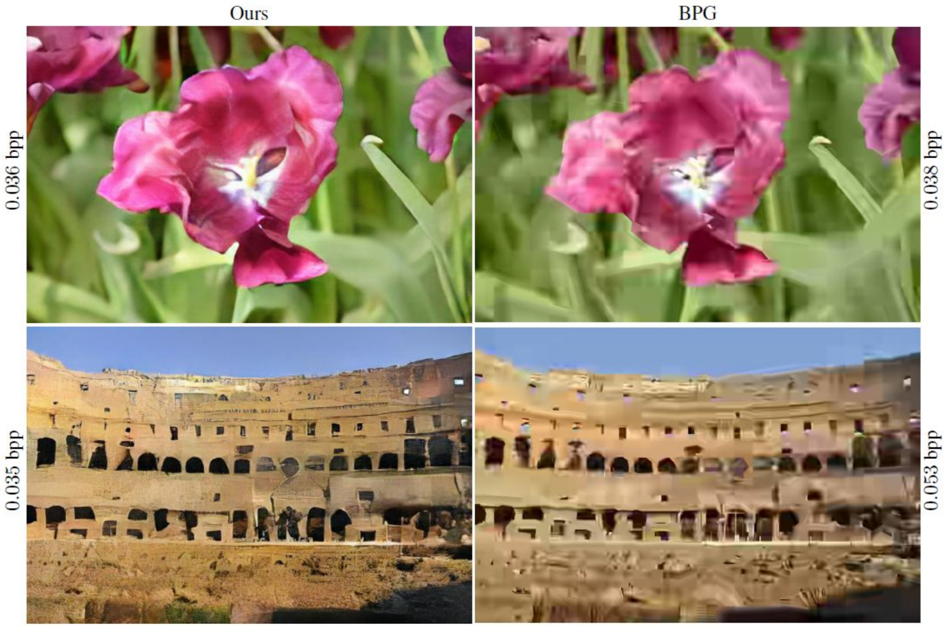
Appendix D.2: Cityscapes Results



Appendix E.1: Compression on Kodak



Appendix E.2: Compression on RAISE1K



Appendix E.3: Comparison with Ripple et al.

Original

Ours, 0.0651bpp

Rippel et al., 0.0840bpp (+29%)

Appendix E.4: Comparison with Minnen et al.

Original

Ours, 0.0685bpp

Minnen et al., 0.155bpp, *127% larger*

BPG, 0.164bpp

Appendix E.4: Comparison with Minnen et al.

Original

Ours, 0.0328bpp

Minnen et al., 0.246bpp, *651% larger*

BPG, 0.248bpp