Perceptual Learned Image Compression With Continuous Rate Adaptation

Shangyin Gao[1], Yibo Shi[1], Tiansheng Guo[1], Zhongying Qiu[1]
Yunying Ge[1], Ze Cui[1], Yihui Feng[1], Jing Wang[1], Bo Bai[1]
[1]Huawei Technologies, Beijing, China
wangjing215@huawei.com

Abstract

In this paper, we propose a perceptual learned image compression framework. We train our networks using rate-distortion, perceptual and adversarial loss in an end-to-end (E2E) manner. To efficiently allocate bits for different image areas, we propose the Region of Interest (RoI) technique in the variable rate adaptation framework. We also investigate the training stability at low bit rate (0.075 bpp) and the superiority of the E2E optimized framework to the post-processing framework. Our proposed framework achieves visually pleasing reconstructions over wide bit-rate range.

1. Introduction

With the increase of images created by ordinary consumers using their smartphones, how to store them efficiently has drawn lots of attention. Lossy image compression uses inexact approximations and partial data discarding to represent the content. In this field, classic image codecs [17] [16] [6] have been developed over the years and achieve good results. The state-of-the-art video compression algorithms [2] can also be used for single image compression.

With the development of deep learning techniques, researchers have also applied deep neural networks to image compression. The learned image compression algorithms take variational autoencoder (VAE) as its basic network and use entropy model for rate estimation [4]. The latest learned image compression algorithms already outperform their classic counterpart in terms of rate-distortion performance. However, deep neural networks optimized only by rate-distortion loss can not fully put its generative capacity into play.

Learned image compression based on Generative Adversarial Networks (GANs) has been studied by lots of researchers [3] [14] [12] [13]. Generative Compression [3] synthesizes details that cannot afford to store and can operate at extremely low bit-rates. HiFiC [14] investigated both network architecture and loss functions and obtains visually pleasing reconstructions at a broad range of bit-rates. The 1st [12] and 2nd [13] ranking methods of CLIC2020 also utilized adversarial training and generated good reconstructions.

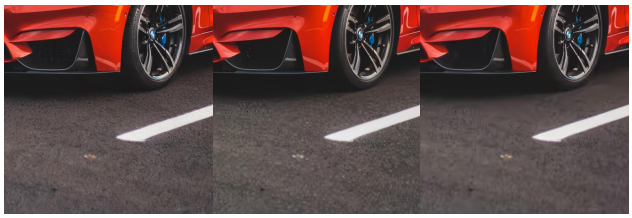

Figure 1. Comparison between reconstructions of our framework (left), winner of CLIC2020 (middle), and VTM at qp 37 (right) at similar bit-rate. *Best viewed on screen.*

Utilizing the most advanced techniques in both learned image compression and GAN fields, we propose the perceptual learned image compression framework. Besides, we propose the RoI method to manually allocate more bit-rate to important image area and achieve good trade-off between bit consumption and image perceptual quality. We also investigate training schemes to stabilize the training at low bit-rate (0.075 bpp) and shows the superiority of E2E optimized framework.

2. Proposed methods

In this section, we present our proposed E2E optimized perceptual image compression framework. We first introduce the detailed network architecture, and the optimization object is then explained. To manually allocate different bits for different image areas, we utilize the RoI technique.

2.1. Architecture

The architecture is shown in Figure 2. Our main architecture includes the encoder, generator, and entropy model. The entropy model includes the hyper-encoder, hyper-decoder, context, and gather. We adversarially train

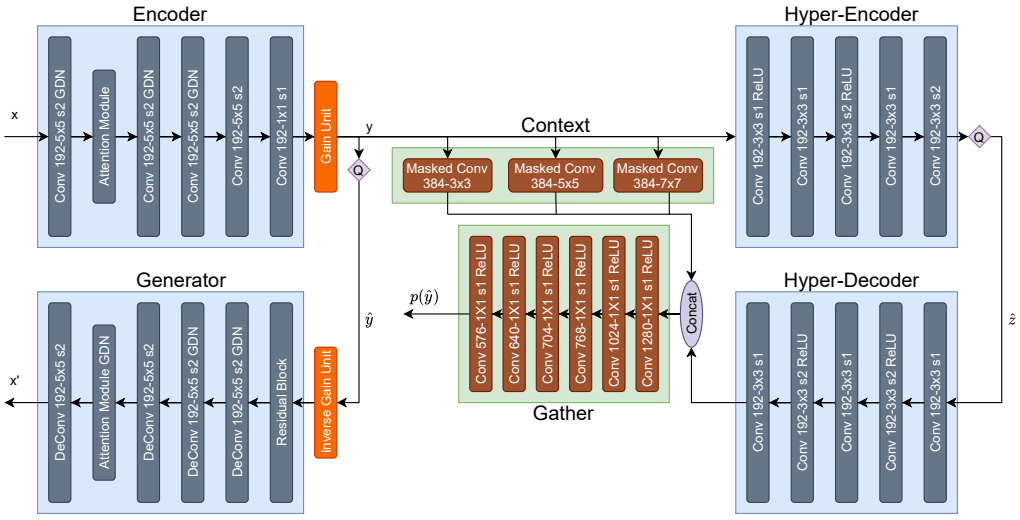

Figure 2. The architecture of our proposed framework. $Conv192\text{-}5\times5$ is a convolution with 192 output channels, with 5x5 filters. $DeConv$ is a deconvolution operation. $s2$ means this convolution or deconvolution is used to downsample or upsample the spatial resolution by ratio 2. GDN or ReLU is used to increase the non-linearity. Gain unit and inverse gain unit are used to achieve continuous rate adaptation.

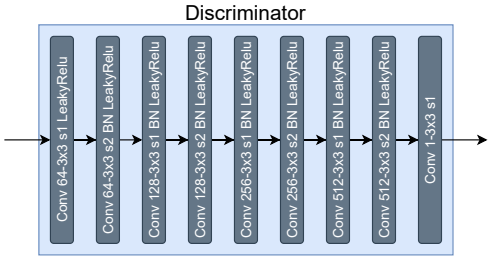

Figure 3. The architecture of discriminator. Same notation is used as in Figure 2.

our framework. The discriminator architecture is also explained in Figure 3.

2.1.1 Autoencoder

In encoder and generator, Generalized Divisive Normalization (GDN) [4] is used to normalize the intermediate feature and add non-linearity. We use the attention module proposed in [7] to increase the capacity of the encoder and generator. As shown in [14], the increase of generator capacity resulting in better performance.

2.1.2 Entropy models

We use the hyper-prior model proposed in [5], where we extract side information z to model the distribution of latent code y and simulate quantization with uniform noise $U(-1/2, 1/2)$ in the hyper-encoder and when estimating $p(\hat{y})$. For the probability estimation of $\hat{z}$, we use a fully factorized density model [5]. $\hat{y}$ is estimated by an asymmetric Gaussian entropy model [8]. The asymmetric entropy model has sufficient degrees of freedom and induces a small estimation error for $\hat{y}$ with the asymmetric distribution. This estimation can be formulated as

$$p(\hat{y}) \sim N(\mu, \sigma_l^2, \sigma_r^2)$$

where σ_l^2 and σ_r^2 represent the left and right scale of a asymmetric Gaussian distribution. All the parameters include μ, σ_l^2 and σ_r^2 are trainable, which increases the computational complexity since the output channel of gather is increased.

2.1.3 Continuous rate adaptation

To achieve flexible rate adaptation in one single model, we also add a pair of gain units [8] [9] between encoder and generator. The gain units pair is used to rescale the magnitude of y. The rescaled y is then quantized. During this process, the gain unit can control the information loss during quantization, therefore, control the bit-rate. The gain unit is made up of a gain matrix $M \in R^{c\times n}$ where c is the channel number of y and n is the number of gain vectors. Each gain vector corresponds to one bit-rate. In our experiments, we always set n as three. After the gain units pair are trained, continuous rate adaptation can be achieved by exponential interpolation between gain vectors without compromising performance[8].

2.1.4 Discriminator

In addition to the basic E2E learned image compression framework described in previous subsections, we also use

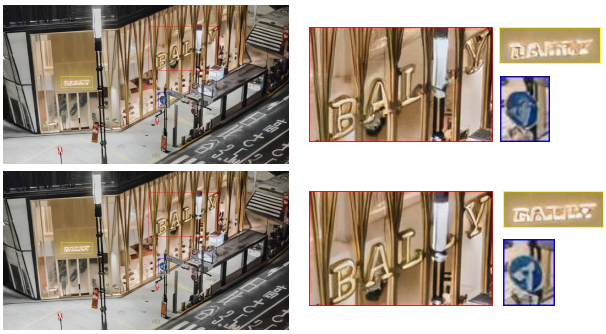

Figure 4. Effectiveness of RoI technique. The image patches in the rectangle are replaced by higher bpp patches. The original version is in the first row and the replaced version is in the second row. The image quality is increased with the cost of bpp increased by 2.7% (from 0.0764 bpp to 0.0785 bpp for the whole image).

adversarial training to fully utilize the generative capacity of the generator and outputs more photo-realistic images. We borrow the discriminator used in ESRGAN [18]. Instead of a standard discriminator, an average relativistic discriminator tries to predict the probability that a real image is relatively more realistic than a fake one. Following the notation in ESRGAN, the discriminator loss is defined as:

$$L_D^{Ra} = -E_{x_r}[log(D_{Ra}(x_r, x_f)] - E_{x_f}[log(1 - D_{Ra}(x_f, x_r)]$$

The adversarial loss for generator is in a symmetrical form:

$$L_G^{Ra} = -E_{x_r}[log(1 - D_{Ra}(x_f, x_r)] - E_{x_f}[log(D_{Ra}(x_r, x_f)]$$

Inspired by pix2pix [11], we enhance the discriminator with the so-called PatchGAN discriminator. The output of D_{Ra} is a matrix instead of a single value. Each value in the matrix corresponds to a patch of the whole input image of the discrimiantor. In this way, the discriminator tries to classify if each patch in an image is real or fake and penalizes the structure at the scale of patches. The PatchGAN can not only retrain more texture but also has fewer parameters and can be applied to arbitrarily large images. The detailed architecture of our discriminator is shown in Figure 3.

2.2. Optimization

The objective of our framework consists of four parts: rate, distortion, perceptual and adversarial loss. For distortion loss, we use L1 loss since it punishes less for large difference compared with Mean Squared Error (MSE) loss. For perceptual loss, we use LPIPS [19] loss instead of VGG Loss, since it alleviates the artifacts [14]. For adversarial loss, we use average relativistic loss, more details are shown in the above section. The overall objective function can be formulated as:

$$L_{total} = \alpha \times L_{rate} + \lambda_1 \times L_{L1} + \lambda_2 \times L_{LPIPS} + \lambda_3 \times L_{GAN}$$

BPP	PNSR	MSSSIM	FID
Low	25.449	0.90502	165.014
Mid	28.329	0.94274	160.297
High	31.015	0.96612	131.711

Table 1. Quantitative results of our method at three bit-rates.

During the training for different target bit-rates, we keep λs relatively fixed and adjust α, which makes our rate constraint much easier than what is used in [14].

2.3. Optimal bit allocation in single image

Lots of professional photographers like to blur the image background to force the viewer to pay more attention to some target objects and increase the aesthetic feeling. Motivated by this, we propose the optimal bit allocation technique to manually allocate more bit on the RoI area and less on the background area. The RoI area can be obtained by manual selection or by using a segmentation framework. This technique can be easily integrated into our framework since our framework uses gain units to achieve continuous rate adaptation in a single model. There exists a spatial correspondence between image x and latent code y. This means, we can allocate fewer bits for the background by decrease the corresponding y through the gain unit, and vice versa, once we got the RoIs in the image. Through this technique, we can decrease the bit consumption while maintaining the image quality or increase the image quality by slightly increase the bit consumption. The effectiveness of RoI technique is shown in Figure 4.

3. Experiments

We implemented our framework using PyTorch. During the training phase, we randomly crop 256x256 patches from the ImageNet dataset and set the batch size to 8. We use Adam optimizer with $\beta_1 = 0.9$ $\beta_2 = 0.999$ to train our networks. The initial learning rate is set to 1e-4 and halved at 160k and 500k iteration. We use kaiming initialization [10] to initialize all our models except for the gain units. The gain units is initialized to one. We set λ_1 to 1e-2 λ_2 to 1 and λ_3 to 5e-4. For different target bit-rates, we only modify α and always use variable-rate models. We set the alpha list to [1.6, 2.6, 4.7] for low bit-rate, [0.75, 1.35, 2.0] for middle bit-rate, and [0.45, 0.6, 1.0] for high bit-rate.

3.1. Stability at low bit-rate

Jointly using GAN with E2E optimized image compression at a low target bit-rate is extremely challenging, since the information used for image reconstruction is very small. In this case, if we still use the same loss weights as in middle and high target bit-rates, the training processing would be unstable and some artifacts will appear. We have three

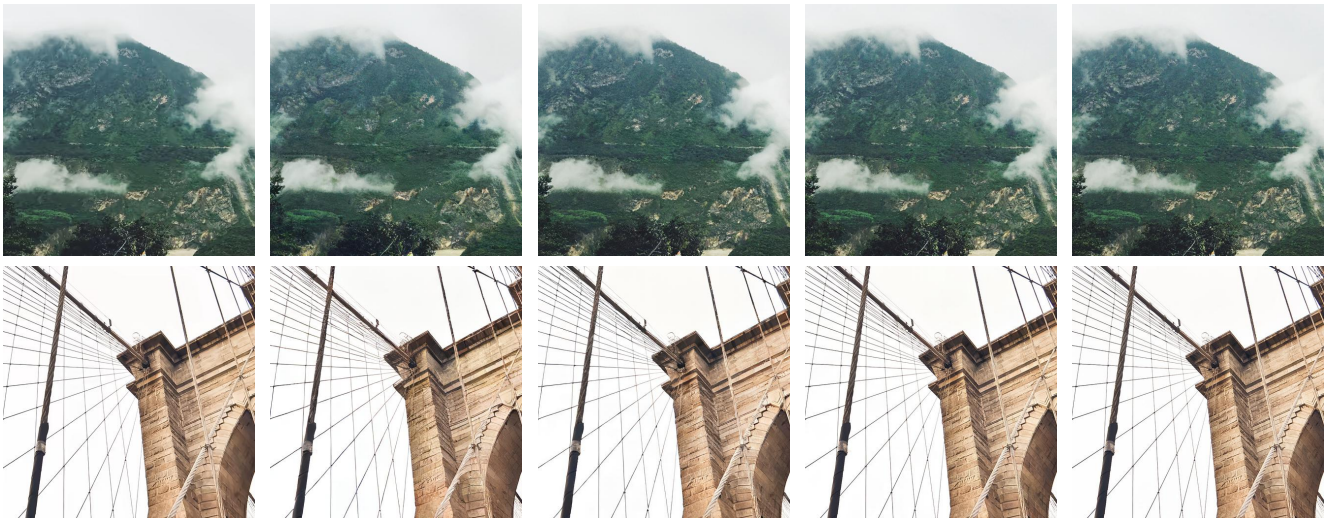

Figure 5. Qualitative comparison of our method at different bit-rates and VTM. 1st column is VTM reconstruction at qp 32 (around 0.3 bpp at CLIC2021 valid dataset). 2nd - 4th column is our method at low (0.075 bpp), mid (0.15 bpp), and high (0.3 bpp) bit-rates. 5th column is the original image patches. *Best viewed on screen.*

options to stabilize the training:

- Increasing the weight for L1 loss term. L1 loss works opposite to adversarial loss, it makes the reconstructed image more pixel-wisely similar to the original image while adversarial loss helps the reconstruction more photo-realistic and stochastic. Therefore, the increment of L1 term aid the training stability. In our experiments, we increase the weight from 0.01 to 0.04.

- Adding average pooling operation after every deconvolution. The most common artifact is the so-called checkerboard artifact. It is well-known that deconvolution layers with non-unit strides cause checkerboard artifacts. We adopt the method proposed in [15] by adding an average pooling layer after deconvolution to avoid the artifact. However, the pooling operation smooths the image and intermediate feature representation. In middle and high target bit-rates, pooling operation is removed.

- Using the variable-rate model to achieve low bit-rate. One advantage of the variable-rate model is that the decoder process both low and high bit-rate information. Experiment shows that the mixed training of different bit-rates improve the performance at the lowest bit-rate.

3.2. Superiority compared with post-processing

Our perceptual image compression framework is E2E optimized. Since all models are jointly trained, the perceptual loss can also guide the information loss of y during quantization. On the contrary, the VTM based post-processing image enhancement method can only use fixed information, therefore, limit the generative ability of deep neural networks. The winner of CLIC2020 is using VTM with post-processing. We compare our method with theirs at the same bit-rates in Figure 1. As can be seen, our framework generates a more realistic road texture. Since the reconstruction of VTM lost detail on the road, the enhancement of road texture is very difficult.

3.3. Qualitative and quantitative results

The qualitative results of our model at three bit-rates are shown in Figure 5. The reconstructions from our method at low bit-rate are more visually pleasing even compared with VTM at high bit-rate. Our reconstructions at all three bit-rates are photo-realistic and without any artifact. With the increase of bit-rate, the images look more similar with original images. We also evaluate our results in terms of PSNR, MSSSIM, and FID on the CLIC2021 valid dataset. The results are shown in Table 1.

4. Conclusion

In this paper, we propose a learned image compression framework oriented for visually pleasing reconstructions. Our framework is optimized by the combination of rate-distortion, perceptual, and adversarial loss and outputs photo-realistic images in a wide bit-rate range. We also investigate the training stability and difference between the E2E optimized and post-processing framework. We also want to implement our framework on MindSpore [1], which is a new deep learning computing framework. These problems are left for future work.

References

[1] Mindspore. *https://www/mindspore.cn*, 2020. 4

[2] Versatile video coding reference software version 10.0. *https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware*$_VTM/ - /releases/VTM - 10.0$, 2020. 1

[3] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative adversarial networks for extreme learned image compression. In *ICCV*, pages 221–231, 2019. 1

[4] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. Density modeling of images using a generalized normalization transformation. *ICLR*, 2016. 1, 2

[5] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *ICLR*, 2018. 2

[6] Fabrice Bellard. Bpg image format. *https://bellard.org/bpg*, 2014. 1

[7] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *CVPR*, 2020. 2

[8] Ze Cui, Jing Wang, Shangyin Gao, Tiansheng Guo, Yihui Feng, and Bo Bai. Asymmetric gained deep image compressionwith continuous rate adaptation. *CVPR*, 2021. 2

[9] TianSheng Guo, Jing Wang, Ze Cui, Yihui Feng Yunying Ge, and Bo Bai. Variable rate image compression with content adaptive optimization. *CVPRW*, 2020. 2

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, pages 1026–1034, 2015. 3

[11] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134, 2017. 3

[12] Younhee Kim, Seunghyun Cho, Jooyoung Lee, Se-Yoon Jeong, Jin Soo Choi, and Jihoon Do. Towards the perceptual quality enhancement of low bit-rate compressed images. In *CVPRW*, pages 136–137, 2020. 1

[13] Jooyoung Lee, Donghyun Kim, Younhee Kim, Hyoungjin Kwon, Jongho Kim, and Taejin Lee. A training method for image compression networks to improve perceptual quality of reconstructions. In *CVPRW*, pages 144–145, 2020. 1

[14] Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. *Advances in Neural Information Processing Systems*, 33, 2020. 1, 2, 3

[15] Yusuke Sugawara, Sayaka Shiota, and Hitoshi Kiya. Super-resolution using convolutional neural networks without any checkerboard artifacts. *ICIP*, 2018. 4

[16] David Taubman and Michael W Marcellin. Jpeg2000 image compression fundamentals, standards and practice. *Journal of Electronic Imaging*, 2013. 1

[17] Gregory K Wallace. The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, 1992. 1

[18] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *ECCV*, September 2018. 3

[19] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 3