Learned Image Compression with Frequency Domain Loss

Soonbin Lee, Jong-Beom Jeong, Inae Kim, and Eun-Seok Ryu
Department of Computer Education, Sungkyunkwan University, Seoul, Republic of South Korea
E-mails: {soonbinlee, uof4949, inaelk, esryu}@skku.edu

***Abstract*—This paper proposes an end-to-end deep image compression model with a frequency domain loss function. Unlike previous deep image compression methods, the model is computed jointly in the frequency domain. By calculating in the frequency domain, the model incorporates high-frequency components to capture detailed information in the reconstructed images effectively. The process of frequency domain relates to the compression technologies, a concept universal to modern image/video codecs (e.g., JPEG), but it has seldom been investigated in a deep image compression model based on neural networks. It was demonstrated that this model shows better image compression performance when measuring visual quality using the peak signal-to-noise ratio, and its rate–distortion performance outperformed traditional neural-network-based models when the model was trained jointly in the frequency domain. This model improves the performance of image compression, especially when the bitrate was low. Moreover, the method can be used and applicable to other compression models easily.**

***Index Terms*—deep learning, image compression**

I. Introduction

Image compression is a well-studied problem in engineering, and it is commonly used to facilitate data transmission and storage. The standard technologies of image compression usually rely on hand-crafted techniques to reduce the spatial redundancy. For example, JPEG makes use of discrete cosine transform (DCT) to transform images from the pixel domain to the frequency domain in the compression process. However, along with the fast development of learning-based image compression methods, the traditional compression methods are not expected to be optimal compression models.

Recently, deep-learning-based image and video compression methods have demonstrated that the image compression task can be performed effectively by deep learning. Using deep-learning methods, promising results have been achieved compared with classical image compression standards. However, previous works have focused on adjusting only the pixel domain in the loss function.

In this study, a model is employed that constructs a frequency domain loss function to compress images efficiently in a way that preserves detailed information, such as high-frequency textures. Unlike previous image compression techniques, this model computes the loss in the frequency domain using DCT [1], which makes it possible to separate high and low-frequency components using the image and video compression codec [2]–[4]. The frequency domain framework makes better compression possible in terms of perceptual quality.

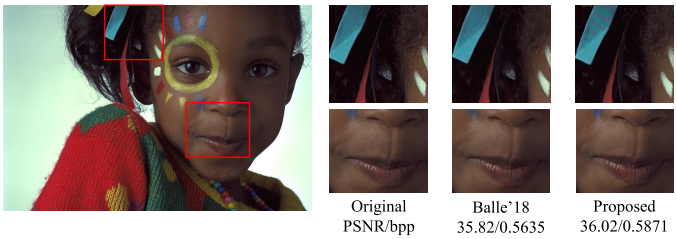

Fig. 1. Example of reconstructed images.

The model uses the frequency domain framework to focus on the most important for reconstruction, by adding the loss to the frequency components in the latent representation. As shown in Fig. 1, the experiments reveal that a model trained with the frequency domain achieves higher quantitative performance — peak signal-to-noise ratio (PSNR)/bits per pixel (bpp) — and preserves more detailed information. The contribution of this study is the introduction of an evaluation with frequency domain loss, which is inspired by the traditional compression codec.

II. Related Work

A. Traditional Codec for Image Compression

In the past few decades, many image and video compression methods have been proposed to improve compression efficiency. The image compression methods mainly focus on reducing the spatial redundancy in images. These methods rely on manually designed techniques, such as block-based coding, quantization, and linear transforms. A typical transform in image compression is to convert the images from the pixel domain to the frequency domain, which is easier for compression. For example, JPEG2000 uses a discrete wavelet transform, and the coefficients are quantized [3].

In video compression, the intra prediction technique is exploited for image compression. For example, the better portable graphics (BPG) standard achieves state-of-the-art image compression performance when compared with previous image codecs, such as JPEG and JPEG2000 [5]. BPG uses 35 encoding intra modes to obtain the residual information, thereby reducing the spatial redundancy. The HEVC/H.265 and versatile video coding (VVC) standards utilize more-advanced techniques for high efficiency coding such as the flexible coding unit (CU) tree size technique [4].

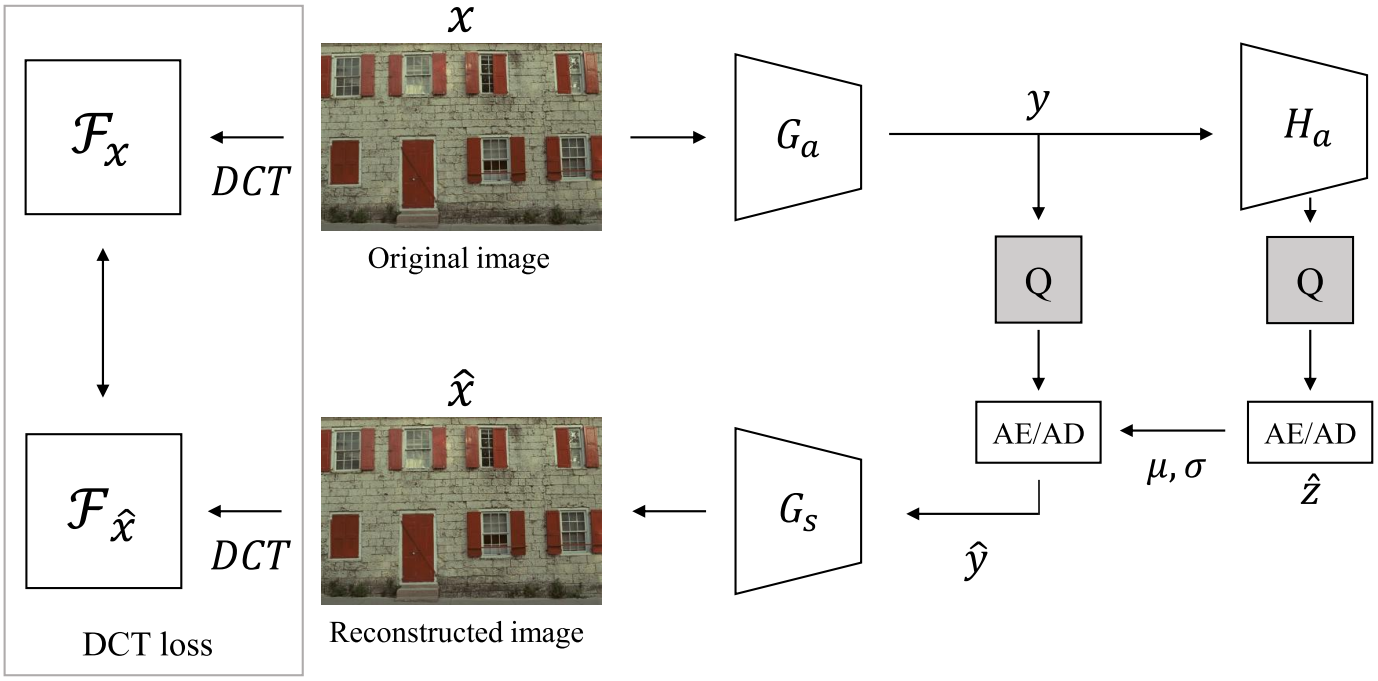

Fig. 2. Architecture of neural network for deep image compression with DCT loss.

B. Learned Image Compression

In the past few years, many studies have been conducted to explore the powerful representation ability of neural networks to enhance image compression. The data-driven learning model to design an efficient compression codec has attracted increasing attention. Most deep image compression models are based on an autoencoder framework. An autoencoder is an encoder that transforms pixels into a latent representation, and a decoder transforms the latent representation back into pixels.

The goal of compression models is to minimize the entropy of the latent representation, which means length of the bitstream, and distortion of the reconstructed image with respect to the original, bringing to a rate–distortion optimization problem:

$$\mathcal{L} = \lambda D + R \tag{1}$$

where λ is the trade-off parameter and D and R represent the distortion and bitrate, respectively.

Several pioneering works have demonstrated that the image compression task can be effectively solved by deep learning. Ballé et al. proposed a convolutional-neural-network-based image compression model by optimizing the rate–distortion in end-to-end schemes [6]. Theis et al. proposed compressive autoencoders for lossy image compression using a Gaussian mixture model [7]. In another study, the continuous relaxation of quantization was proposed to solve the zero-gradient problem in another way [8]. The model presented by Ballé et al., which has a fully factorized prior, was extended in another study with a hyperprior that captures the information of spatially adjacent pixels [9]–[11]. [10] introduced an adaptive entropy model that estimates the scale of the distribution for each representation. They assumed that the scales of the latent representations from the input images vary within a neighboring area. In that study, the hyperencoder and hyperdecoder modules were used to estimate the hyperprior information for the entropy model. Specifically, in the latent representation y, the hyperencoder module calculates the scale of distribution and encodes it to latent representations z. Then the latent representations z are quantized as $\hat{z}$, and $\hat{z}$ is sent by arithmetic coding. Finally, The entire learning-based image compression framework is optimized by considering the rate-distortion trade-off in the following way:

$$\mathcal{L} = \lambda D + R = \lambda d(x, \hat{x}) + H(\hat{y}) + H(\hat{z}) \tag{2}$$

Here, $d(\,\cdot\,)$ is the distortion metric such as the PSNR and perceptual loss and H represents the bitrate for encoding latent representations y and z [12]. In the previous study, the bitrate was approximated by using the entropy of the corresponding latent representations in the following way:

$$H(\hat{y}) = E\left[-\log_2\left(p_{\hat{y}|\hat{z}}(\hat{y} \mid \hat{z})\right)\right] \tag{3}$$

$$H(\hat{z}) = E\left[-\log_2\left(p_{\hat{z}}(\hat{z})\right)\right] \tag{4}$$

Here, $p_{\hat{y}|\hat{z}}(\hat{y} \mid \hat{z})$ and $p_{\hat{z}}(\hat{z})$ represent the distributions of $\hat{y}$ and $\hat{z}$, respectively. The main experiment in this work was conducted to compare the performance of the hyperprior model that is optimized for mean squared error (MSE) pixel loss [10].

III. Methodology

In this section, deep image compression with frequency domain loss is introduced. In the experiment, the DCT, which is a Fourier transform that transforms a signal into the frequency domain, was used. The main characteristic of the DCT is the use of only cosine wave bases, which is critical for compression because fewer cosine functions are needed to approximate the input signal. The 2D DCT is formally defined as follows [1]:

$$\mathcal{F}\left(\mathbf{X}_{jk}\right) =$$

$$\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} x_{mn} \cos\left(\pi \frac{j}{M}\left(n+\frac{1}{2}\right)\right) \cos\left(\pi \frac{k}{N}\left(m+\frac{1}{2}\right)\right)$$

where $\mathbf{X}$ is input image patches, M and N are the height and width of the image patches, and m and n are the usual row and column indices so that x_{mn} is the value at row m column $n, 0 \leq j \leq M-1$, and $0 \leq k \leq N-1$

$$\mathcal{L}_{\text{DCT}} = \left\|\mathcal{F}(x) - \mathcal{F}(\hat{x})\right\|_2 \tag{5}$$

In the experiment, the 2D DCT function was computed over 8×8 nonoverlapping patches of the given image, and $\hat{x}$ and $\mathcal{F}$ represent the reconstructed image and DCT function, respectively.

$$\mathcal{L}_{Total} = \mathcal{L} + \alpha \mathcal{L}_{\text{DCT}} \tag{6}$$

Then the DCT loss is trained jointly to focus on frequency components, according to their relative importance to detailed information.

IV. Implementation details

Experiments were performed to demonstrate the effectiveness of the proposed method, which is trained with DCT loss. Basically, the same training strategy was followed as that in a previous study [13]. In this experiment, 20,577 original *Flicker* images from Flick.com were used, and 256 × 256 cropped patches were randomly taken for training. For performance evaluation, all images in the Kodak PhotoCD image dataset were used as a test dataset [14].

For image compression in various bitrates, the model was trained using different λ values (i.e., 256, 512, 1024, 2048, 4096, 6144, and 8192). First, the model optimized by the MSE loss function with λ = 8192 was adopted as the pretrained model. For other bitrates, the model trained on a high bitrate (λ = 8192) and low bitrate (λ = 2048) was adopted as a pretrained model. The model was trained on 3 GPU (NVIDIA 2080Ti) with a batch size of 8. In addition, N was set to 192 and M to 320 for the high bitrate and N to 128 and M to 192 for the low bitrate.

The Adam optimizer was used with a learning rate of 1×10^{-4} in the first 1,000,000 iterations and 1×10^{-5} in the remaining 500,000 iterations [15]. After 1,500,000 iterations,the MSE loss function was changed to the DCT loss function, and the pretrained model was fine-tuned with different α values and a batch size of 96. Then, the model was trained jointly with a learning rate of 1×10^{-5} for 50,000 iterations.

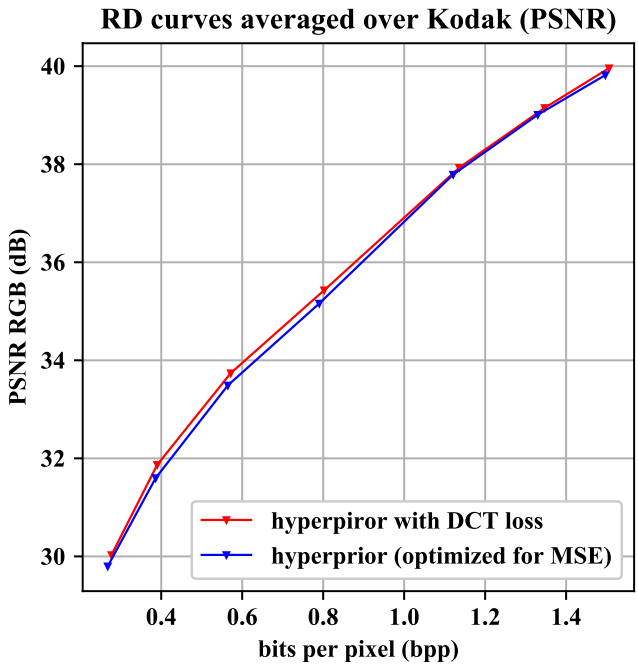

Fig. 3. Rate–distortion curves of the proposed method when compared to those in a previous study [10], optimized for pixel loss (MSE).

V. Experimental Results

The method was compared with that in a previous study [10], a hyperprior model with only pixel loss. As shown in Fig. 3, the proposed model had slightly better image compression performance, especially for low bit rates. Although DCT loss was not optimized for pixel loss, the proposed model showed a gain in the PSNR metric. The proposed method was also compared with well-known image compression standards, such as BPG, JPEG, JPEG2000, and deep-learning-based methods [9], [10]. Fig. 4 illustrates the results. The proposed model performed better than that in the previous study, but it did not surpass the BPG 4:4:4 compression model.

TABLE I
Average PSNR and BPP achieved with DCT loss when tranined by pixel loss (MSE)

lambda(λ)	w/o DCT loss (PSNR, bpp)	DCT loss (PSNR, bpp)
256	(29.80, 0.267)	(**30.03**, 0.277)
512	(31.60, 0.386)	(**31.87**, 0.390)
1024	(33.49, 0.564)	(**33.74**, 0.571)
2048	(35.16, 0.790)	(**35.43**, 0.812)
4096	(37.79, 1.121)	(**37.93**, 1.136)
6144	(39.01, 1.330)	(**39.15**, 1.347)
8192	(39.82, 1.497)	(**39.96**, 1.506)

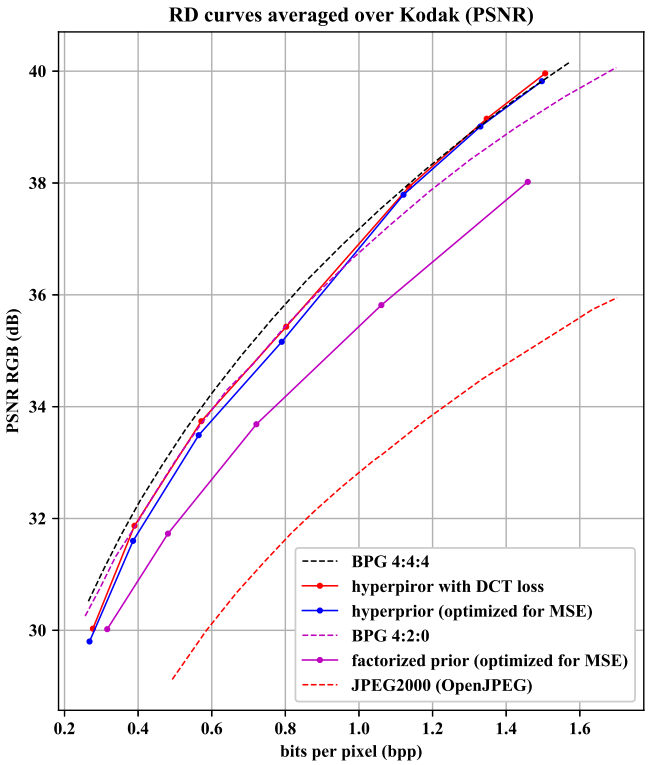

Fig. 4. Rate–distortion curves of the proposed method and other methods for image compression when using PSNR metric.

VI. Conclusion

The frequency domain loss function was used for training image neural networks for end-to-end image compression. Modest improvements in quantitative performance were achieved, especially when the bitrate was low. The main limitation of this method because of the complexity of DCT is the burden in training time. Future work should explore the performance of frequency domain loss in image compression for other metrics, such as multi-scale structural similarity (MS-SSIM) metric.

Acknowledgment

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2019R1A2C1010476).

References

[1] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. C-23, no. 1, pp. 90–93, 1974.

[2] G. K. Wallace, "The jpeg still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[3] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The jpeg 2000 still image compression standard," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 36–58, 2001.

[4] G. J. Sullivan, J. Ohm, W. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[5] "F. bellard, bpg image format." http://bellard.org/bpg/, accessed: 2020-10-30.

[6] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," *arXiv preprint arXiv:1611.01704*, 2016.

[7] L. Theis, W. Shi, A. Cunningham, and F. Huszár, "Lossy image compression with compressive autoencoders," *arXiv preprint arXiv:1703.00395*, 2017.

[8] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, and L. Van Gool, "Soft-to-hard vector quantization for end-to-end learned compression of images and neural networks," *arXiv preprint arXiv:1704.00648*, vol. 3, 2017.

[9] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," *Advances in Neural Information Processing Systems*, vol. 31, pp. 10 771–10 780, 2018.

[10] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *arXiv preprint arXiv:1802.01436*, 2018.

[11] J. Lee, S. Cho, and S.-K. Beack, "Context-adaptive entropy model for end-to-end optimized image compression," *arXiv preprint arXiv:1809.10452*, 2018.

[12] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*. Springer, 2016, pp. 694–711.

[13] J. Liu, G. Lu, Z. Hu, and D. Xu, "A unified end-to-end framework for efficient deep image compression," *arXiv preprint arXiv:2002.03370*, 2020.

[14] "E. kodak, kodak lossless true color image suite (photocd pcd0992). [online]." http://r0k.us/graphics/kodak/.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.