An Enhanced Multi-frequency Learned Image Compression Method

Lin He, Zhihui Wei(✉), Yang Xu, and Zebin Wu

Nanjing University of Science and Technology, Nanjing, China
gswei@njust.edu.cn

Abstract. Learned image compression methods have represented the potential to outperform the traditional image compression methods in recent times. However, current learned image compression methods utilize the same spatial resolution for latent variables, which contains some redundancies. By representing different frequency latent variables with different spatial resolutions, the spatial redundancy is reduced, which improves the R-D performance. Based on the recently introduced generalized octave convolutions, which factorize latent variables into different frequency components, an enhanced multi-frequency learned image compression method is introduced. In this paper, we incorporate the channel attention module into multi-frequency learned image compression network to improve the performance of adaptive code word assignment. By using the attention module to capture the global correlation of latent variables, complex parts of the image such as textures and boundaries can be better reconstructed. Besides, an enhancement module on decoder side is utilized to generate gains. Our method shows the great visual appearance and achieves a better grade on the MS-SSIM distortion metrics at low bit rates than other standard codecs and learning-based image compression methods.

Keywords: Learned image compression · Multi-frequency image coding · Channel attention · Decoder enhancement

1 Introduction

In the 5G era, smart terminals will see a new round of explosive growth, in which image data growth being particularly prominent. It becomes important to obtain satisfactorily compressed images based on limited hardware resources. Instead of saving the original RGB data, a lossy version of the image is stored that is as close as possible to the original in terms of visual experience. Traditional image compression methods [6,21,23,26,30] are usually composed of transform, quantization, and entropy coding, which rely on manual optimization of each module. However, hand-crafted tuning of each module may not lead to an overall performance improvement, which may limit their performance. In

© Springer Nature Switzerland AG 2021
H. Ma et al. (Eds.): PRCV 2021, LNCS 13021, pp. 189–200, 2021.
https://doi.org/10.1007/978-3-030-88010-1_16

the meanwhile, learned image compression methods [1–5,8,9,12,13,15,16,18–20,24,25,27–29] has attracted more and more attention. The advanced learning-based image compression methods have achieved superior performance over traditional methods(i.e. BPG [6]).

The lossy image compression methods are commonly improved in two ways: designing more efficient transformations and building more refined probabilistic statistical models of latent variables. It is experimentally demonstrated that the GDN layer can effectively Gaussianize the local joint statistical information of natural images, thus achieving local independence to a certain extent. However, latent variables are usually expressed by feature maps with no differentiation in spatial resolution, where exits some spatial redundancies. This indicates that better R-D performance can be achieved by feature maps with spatial resolution, which reduce the spatial redundancy. In [7], octave convolutions are utilized to decompose the latent variables into high-frequency and low-frequency factors . In [3], the generalized octave convolutions are proposed to accommodate image compression applications.

For the entropy modeling, in [4], the latent representations are modeled as independently identically distribution across space and channels, then in [5], the entropy model is conditional Gaussian scale mixture(GSM) and codes are modeled as conditionally independent given the hyper-prior. Most recent learned image compression techniques utilize the context-adaptive entropy method, which combines the super-priority and autoregressive models [20].

In this paper, the idea of multiple spatial frequencies is adopted, based on generalized octave convolution, a multi-frequency channel attention module is introduced to improve coding performance. Besides, the enhancement module is introduced on decoder side to enhance compression. Better image compression performance especially at a low bit rate is obtained when compared with recently advanced image compression methods.

The contributions of this paper are generalized as follows:

- We combine the channel attention technique with multi-frequency potential representations to improve coding performance.
- We apply an enhancement module on the decoder side for further compression enhancement.
- The proposed framework obtains better image compression performance at a low bit rate compared to other recently advanced methods [3,6,8,20].

2 Related Works

Many image compression methods have been developed and some standards have been successfully established over the past decades, such as JPEG [30], JPEG2000 [23], and BPG [6]. But these existing methods rely on hand-crafted modules, which include transform, quantization, and entropy coding such as Huffman encoder. Recently, the internal prediction technique which is firstly used in video compression has also been utilized for image compression. For example, as the recently advanced technique compared with other manually

designed techniques, the BPG [6] standard is based on the video compression standard HEVC/H.265 [26], which adopts the prediction-transform technique to reduce the spatial redundancy.

In learned image compression methods, some works design efficient network structures to extract more compact latent variables and rebuild high-quality images from compressed features. For example, [13,28,29] utilize recurrent networks to compress the residual signal progressively, and achieve scalable coding with binary representation in each recurrence. Although recurrent models can handle variable bit rates compression naturally, they usually take more time for encoding and decoding because the network is executed multiple times. [1,4,5,15,16,19,27] utilize full convolutional networks, which are trained under rate-distortion constraints, with a trade-off between bit rate and reconstruction quality. Since each trained model corresponds to a Lagrangian coefficient λ which controls balance, multiple models should be trained to fit the needs of variable bit rates.

Some works establish a probability model for effective entropy coding, the modeled objects of the entropy model are divided into binary code streams and latent variables. The binary code stream is the output of the encoder, and the bitstream allocation is guided directly by modeling the binary code stream with an entropy model. The latent variables are the output of the analysis transform, which is generally a real number representation. The difference between binary code streams and latent variables is that binary code streams are the final output of the encoder, while the latent variables need to be quantized and entropy coded to obtain the bitstream which is the final output of the encoder. By modeling the entropy of the latent variables and adding end-to-end optimization, facilitates the coding network to generate a tightly compressed representation.

Based on the binary code streams, Toderici et al. [29] added an entropy model in subsequent work and used PixelRNN [22] to estimate the probability distribution of all symbols to be encoded based on the encoded symbols. Because there exist spatial redundancies in natural images, predicting the probability or coding residuals of the current target based on the context can improve the image compression performance. In addition to using contextual information to guide the encoding of the current target, Covell et al. [9] used a mask based on RNN to guide the symbol assignment, which allows compression system to adaptively change the number of bits transmitted based on the local content. Li et al. [16] used Convolutional Neural Networks (CNNs) to learn importance maps and constructed masks to indicate the length of binary codes.

Based on the latent variables, Balle [4] approximates the actual probability distribution of the symbols using a segmented linear model, with the latent variables first undergoing a GDN transformation that greatly reduces the spatial redundancy between pixels to achieve a factorized probability model of the latent variables; Agustsson [1] estimates the probability distribution of the symbols by their histograms; Theis [27] uses Laplace smoothing histograms to better estimate the probability distributions. All these models focus on learning the distribution of the representation without considering adaptivity, in other words,

once the entropy model is trained, the parameters of the trained model are fixed for any input during testing. There exit large spatial dependencies in the quantified latent variables. The standard approach to modeling dependencies among a set of target variables is to introduce hidden variables, conditioned on the assumption that the target variables are independent. Balle [5] proposed a hyper-prior entropy model that introduces an additional set of random variables to capture spatial dependencies. Minnen [20] added masked convolution as a contextual model for autoregression to more fully exploit the domain relevance of the predicted pixels. Lee [15] used two types of contextual models to estimate the Gaussian distribution for each latent variable. Cheng [8] models the distribution of latent variables as a discrete Gaussian mixture model(GMM) and adds a simplified version of the Attention module, which enables the learning model to focus more on complex regions. Liu [18] adds an enhancement module at the decoder side. Hu [12] proposes a framework with a superior hierarchical framework with multi-layer superiority representation. Some methods [2,24,25] utilize generative models to learn the distribution of input signals and generalize subjectively excellent reconstructed image at extremely low bit rates.

3 Proposed Method

3.1 Formulation of Multi-frequency Learned Compression Models

The architecture of the whole scheme discussed in this paper is shown in Fig. 1. Inspired by recent advances in learned image compression [4,5,20], an autoencoder style network is performed. Specifically, the generalized octave convolutions [3] shown in Fig. 2 are utilized to reduce spatial redundancy which improves the R-D performance. The entire framework consists of five main modules, which are encoder network, decoder network, hyper encoder network, hyper decoder network, and parameter estimator network.

The encoder network transforms the original image x into the corresponding latent variables y. Since the generalized octave convolutions are utilized, the latent variables are decomposed into high-frequency (HF) and low-frequency (LF) factors(denoted by y^H and y^L), where the lower frequency corresponds to low spatial resolution. The internal structure of generalized octave convolution and the corresponding transposed structure is shown in Fig. 2. To further reduce spatial redundancy, channel attention modules are applied separately on y^H and y^L, then y^H_{at} and y^L_{at} are obtained. The latent variables y^H_{at} and y^L_{at} will be quantized to $\tilde{y}^H_{at}$ and $\tilde{y}^L_{at}$. The next part is the arithmetic encoder and arithmetic decoder, where is considered as lossless entropy coding. Then the quantized latent variables $\tilde{y}^H$ and $\tilde{y}^L$ were fed to the decoder network to obtain the reconstructed image $\tilde{x}$. In this paper, the quantization strategy is the same as [4].

The hyper encoder, hyper decoder, and params estimator modules are utilized for estimation of the distribution of latent variables. Since the image compression methods aim to obtain a reconstructed image at a given bit rate, an accurate entropy model which estimates the bit rate is critical. The whole

pipeline is like [20], based on the latent variables, a context model and a hyper auto-encoder are combined to exploit the probabilistic structure. The context model is a kind of autoregressive model for latent variables with different resolutions, which corrects the prediction based on the previous content. The results of the context model are denoted as ϕ^H and ϕ^L. The hyper encoder module is utilized to represent side information efficiently and encode side information into latent variables. The results of the hyper autoencoder are denoted as ψ^H and ψ^L. Since the generalized octave convolution is utilized in hyper autoencoder, we get high and low frequency latent variables z^H and z^L. Channel attention modules are applied separately on z^H and z^L. Similar to previous operations, z_{at}^H and z_{at}^L are quantized into $\tilde{z}^H$ and $\tilde{z}^L$, then sent by arithmetic coding. The statistical model of $\tilde{y}^H$ and $\tilde{y}^L$ is assumed to be conditional Gaussian entropy model [20]. To estimate the means and standard deviations of conditional Gaussian distributions for each latent variables, the params estimator module utilize the outputs of both context model (ϕ^H,ϕ^L) and hyper decoder(ψ^H, ψ^L) for better performance.

The learned image compression network is optimized by trade-off between code rate and distortion. Rate (R) is the estimated number of consumed bits after arithmetic encoding, while distortion (D) is loss of reconstructed images. We utilize a Lagrange multiplier λ as the trade-off parameter. The loss function is written as:

$$\begin{aligned} L &= R + \lambda D \\ &= R^H + R^L + \lambda d(x, \tilde{x}) \end{aligned} \tag{1}$$

where R^H and R^L are seperately the rates of high-frequency and low-frequency latent variables, which can be defined as:

$$\begin{aligned} R^H &= H(\tilde{y}^H) + H(\tilde{z}^H) \\ &= E(-log_2(p_{\tilde{y}^H|\tilde{z}^H}(\tilde{y}^H|\tilde{z}^H))) + E[-log_2(p_{\tilde{z}^H}(\tilde{z}^H))] \\ R^L &= H(\tilde{y}^L) + H(\tilde{z}^L) \\ &= E(-log_2(p_{\tilde{y}^L|\tilde{z}^L}(\tilde{y}^L|\tilde{z}^L))) + E[-log_2(p_{\tilde{z}^L}(\tilde{z}^L))] \end{aligned} \tag{2}$$

the $p_{\tilde{y}^H|\tilde{z}^H}$ and $p_{\tilde{y}^L|\tilde{z}^L}$ are respectively the conditional Gaussian entropy model for high-frequency and low-frequency latent variables. Besides, the mean and scale parameters μ_i^H , σ_i^H, μ_i^L and σ_i^L are obtained by params estimator module f_pe^H and f_pe^L. Then the distribution of the latent variables can be formulated as:

$$\begin{aligned} p_{\tilde{y}^H|\tilde{z}^H}(\tilde{y}^H|\tilde{z}^H) &= \prod_i (\mathcal{N}(\mu_i^H, {\sigma_i^2}^H) * \mathcal{U}(-\frac{1}{2}, \frac{1}{2}))(\tilde{y}_i^H) \\ p_{\tilde{y}^L|\tilde{z}^L}(\tilde{y}^L|\tilde{z}^L) &= \prod_i (\mathcal{N}(\mu_i^L, {\sigma_i^2}^L) * \mathcal{U}(-\frac{1}{2}, \frac{1}{2}))(\tilde{y}_i^L) \end{aligned} \tag{3}$$

the $p_{\tilde{z}^H}$ and $p_{\tilde{z}^L}$ are supposed to be independent and identically distributed(i.i.d), and a non-parametric factorized model is utilized [4]. Then the distribution of the latent variables can be formulated as:

$$p_{\tilde{z}^H|\Theta^H}(\tilde{z}^H|\Theta^H) = \prod_j (p_{\tilde{z}_i^H|\Theta_j^H}(\Theta_j^H) * \mathcal{U}(-\frac{1}{2}, \frac{1}{2}))(\tilde{z}_j^H)$$
$$p_{\tilde{z}^L|\Theta^L}(\tilde{z}^L|\Theta^L) = \prod_j (p_{\tilde{z}_i^L|\Theta_j^L}(\Theta_j^L) * \mathcal{U}(-\frac{1}{2}, \frac{1}{2}))(\tilde{z}_j^L) \quad (4)$$

where Θ^H and Θ^L denote the parameter vectors.

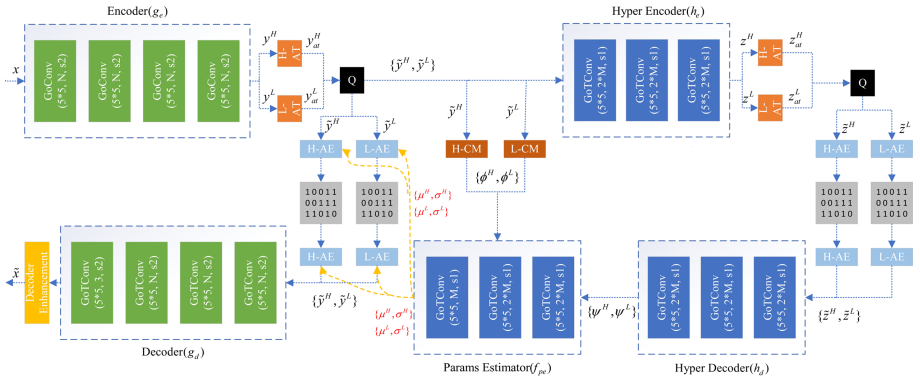
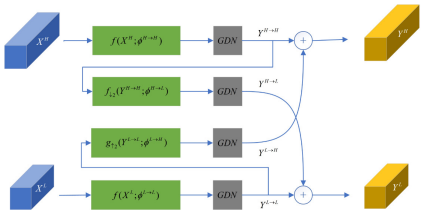
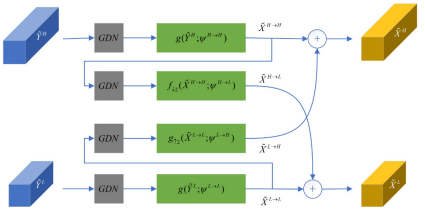


Fig. 1. The overall framework of the proposed learned image compression method. **H-AT** and **L-AT**: attention modules for HF and LF latent variables. **H-AE** and **H-AD**: arithmetic encoder and decoder for HF latent variables. **L-AE** and **L-AD**: arithmetic encoder and decoder for LF latent variables. **H-CM** and **L-CM**: context models for HF and LF latent variables, composed of one 5 * 5 masked convolution layer. **Q**: quantization [4]



(a) GoConv. X^H, X^L: input HF and LF feature maps; Y^H, Y^L: output HF and LF feature maps



(b) GoTConv. $\tilde{Y}^H, \tilde{Y}^L$: input HF and LF feature maps; $\tilde{X}^H, \tilde{X}^L$: output HF and LF feature maps

Fig. 2. Architecture of the generalized octave convolution (GoConv) and transposed-convolution(GoTconv) [3]. GDN: the activation layer [4]; f: regular convolution; $f_{\downarrow 2}$: regular convolution with stride 2; g: regular transposed convolution; $g_{\uparrow 2}$: regular transposed convolution with stride 2.

3.2 Channel Attention Scheme

Some works have utilized spatial attention mechanisms to reduce spatial redundancy [8,18]. Inspired by [11], a channel attention model is proposed to focus on channels of outstanding local importance, and then reduce spatial redundancy. The structure of the channel attention module for latent variable with different spatial resolutions is shown in Fig. 3. By using the attention module to capture the global correlation of latent variables, complex parts of the image such as textures and boundariescan be better reconstructed.

For a feature map $X \in R^{h \times w \times c}$, first, a global average pooling is utilized to achieve statistical channel importance $t \in R^c$:

$$t_c = \frac{1}{h \times w} \sum_{i=1}^{h} \sum_{j=1}^{w} x_c(i,j) \tag{5}$$

where $x_c(i,j)$ is the value at the (x,y) position in $c-th$ channel of feature map X. Then, several non-linear transforms are utilized to capture the channel-wise relationship, which can be denoted as:

$$s = \sigma(F_2 \delta(F_1 t)) \tag{6}$$

where F_2 and F_1 are the fully connected layers, δ is the ReLU activation function, σ is the sigmoid function. Finally, by rescaling the feature map X with s and adding the residual operation, a feature map with channel attention applied is obtained.

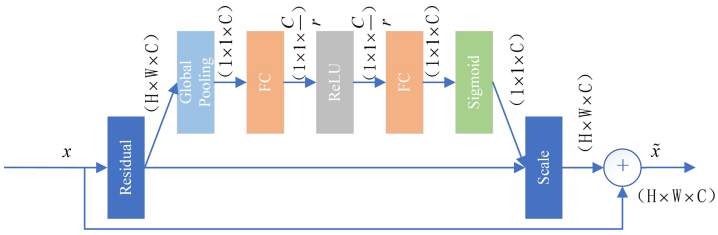


Fig. 3. Channel attention module on both HF and LF feature map. FC: fully connected layer. r is chosen to be 16.

3.3 Decoder-Side Enhancement

To further enhance the quality of the reconstructed images, an enhancement module at the decoder side is introduced. Influenced by image super-resolution solutions [17], we utilize the residual block to further improve image quality.

It has been experimentally proved that the residual blocks also work in super-resolution problems. But the original ResNet was originally proposed to solve

problems such as classification and detection. Since the batch normalization layer consumes the same size of memory as the convolutional layer before it, after removing this step of operation, we can stack more network layers or make more features extracted per layer, thus getting better performance. In each residual block, a constant scaling layer is placed after the final convolutional layer. These blocks greatly stabilize the training process when a large number of filters are used. The structure of the decoder-side enhancement module is shown in Fig. 4.

First, to expand the channel dimensions from 3 to 32, a convolutional layer is utilized. Then, three enhancement blocks are applied, each contains three residual blocks where remove the batch normalization operation to keep details and save computational resources. Finally, a convolution layer is applied to transfer channel dimension to 3 and apply the residual operation to get the reconstructed image.

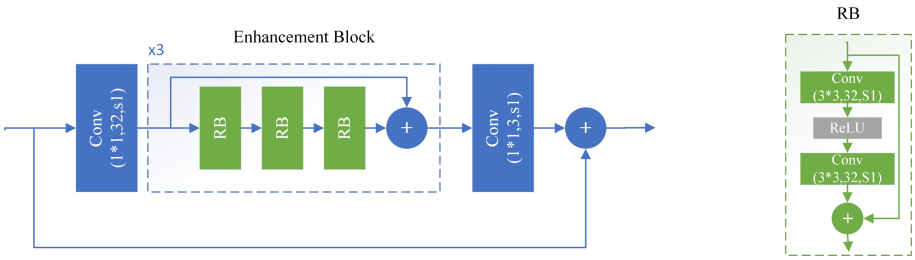


Fig. 4. Enhancement module on the decoder side. RB: the residual block

4 Experiment Results

A partial subset of ImageNet [10] is selected for training, which contains 6080 pictures in total. The size of the images is first randomly cropped into $256 \times 256 \times 3$ patches. Pixel values are normalized from (0,255) to (0,1). The standard Kodak dataset [14] is utilized for testing, which contains 24 high-quality uncompressed PNG images.

4.1 Parameter Description

In the encoder module(Fig. 1), output channel size $M = 192$, the sizes of HF and LF latent variables (y^H and y^L) are respectively $16 \times 16 \times 96$ and $8 \times 8 \times 96$. The ratio of LF is 0.5, which means that half of the latent representations are assumed to be LF part. All modules are jointly trained over 100 epochs with Adam solver, with the batch size set as 16 and the learning rate fixed at 0.001. The trade-off parameter λ takes values from the range [0.0005,0.1].

4.2 Results Evaluation

The compression capability of the proposed framework is compared with the traditional image compression methods including JPEG2000 [23], WebP, BPG(4:2:0) [6], and also recently advanced learning-based image compression methods [3,5,8,20]. We utilize MS-SSIM as the evaluation indicators, which is more consistent with human eye visual perception than other evaluation metrics like PSNR and SSIM. The comparison result on the Kodak dataset is shown in Fig. 5, which is an average result over 24 images. The R-D curve is plotted based on multiple bpp points, which are corresponding to different bit rates. Several models are trained with different values of λ to achieve different bit rates.

As shown in Fig. 5, the proposed scheme outperforms the standard codecs and most advanced methods at the low bit rates(bpp < 0.25). Compared with the recently advanced standard codecs, such as BPG (4:2:0) [6], the proposed method achieves better performance at each bit rate. Some visual examples for visualization details is shown in Fig. 6 and Fig. 7. As seen in the example, at a low bit rate, our method behaves the best compared to the others, the high-frequency details like eyelashes and curly hair are clearly expressed. While the image reconstructed by JPEG [30] faces the problems of artifacts. In particular, our method has a higher MS-SSIM score at the low bpp points compared to original framework based on GoConve [3] and Balle's method [5].

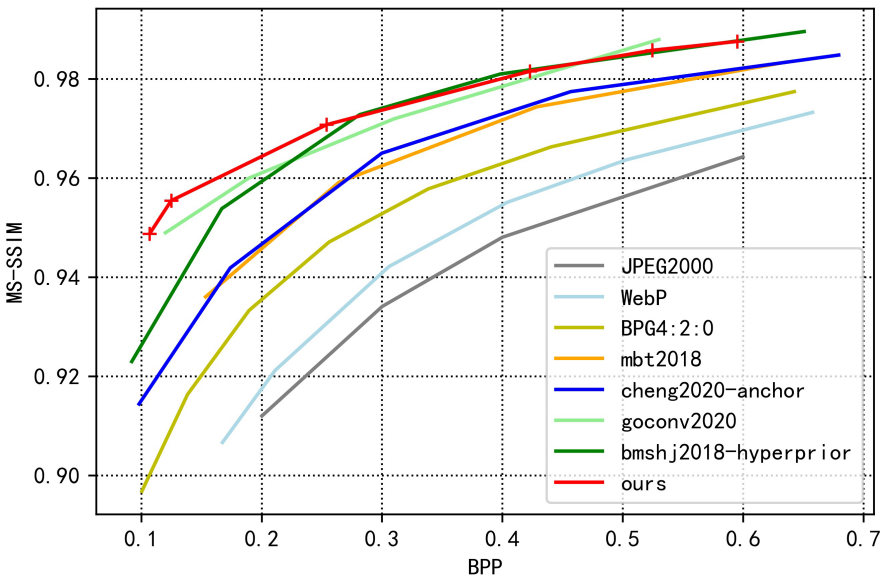
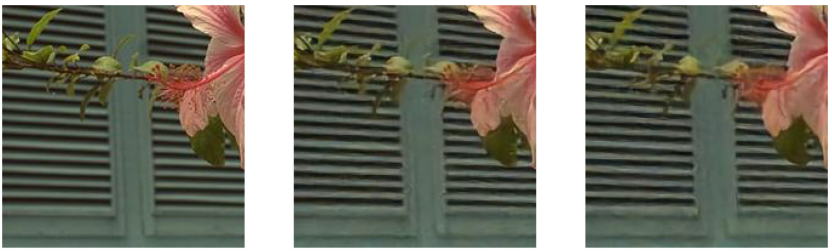


Fig. 5. Kodak comparison results.

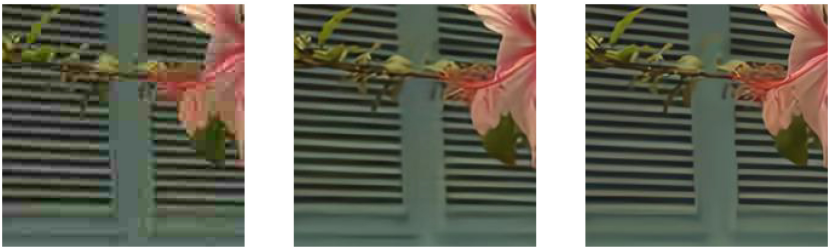
(a) original image (b) 0.945/0.111bpp (c) 0.939/0.112bpp

(d) 0.712/0.095bpp (e) 0.927/0.144bpp (f) 0.933/0.106bpp

Fig. 6. Visualization of partial reconstructed kodim04 from Kodak dataset, **our proposed method (b)**, GoConv2020 (c), JPEG (d), Balle et al. 2018 (e), and Cheng2020-anchor (f). We take MS-SSIM as the metrics.



(a) original image (b) 0.974/0.114bpp (c) 0.966/0.112bpp



(d) 0.836/0.122bpp (e) 0.969/0.179bpp (f) 0.972/0.142bpp

Fig. 7. Visualization of partial reconstructed kodim07 from Kodak dataset, **our proposed method (b)**, GoConv2020 (c), JPEG (d), Balle et al. (2018) (e), and Cheng2020-anchor (f). We take MS-SSIM as the metrics.

5 Conclusion

We propose an enhanced multi-frequency learned image compression framework in this paper. Using the generalized octave convolution, the latent variables are divided into high-frequency and low-frequency components, while the high frequency part is represented by a higher spatial resolution. The channel attention modules for high-frequency and low-frequency latent variables are proposed to further reduce spatial redundancy. Finally, an enhancement module on decoder side is utilized to further enhance performance. The whole framework is trained end to end and achieves better performance at a low bite rate compared with recently advanced methods.

References

1. Agustsson, E., et al.: Soft-to-hard vector quantization for end-to-end learning compressible representations. arXiv preprint arXiv:1704.00648 (2017)
2. Agustsson, E., Tschannen, M., Mentzer, F., Timofte, R., Gool, L.V.: Generative adversarial networks for extreme learned image compression. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 221–231 (2019)
3. Akbari, M., Liang, J., Han, J., Tu, C.: Generalized octave convolutions for learned multi-frequency image compression. arXiv preprint arXiv:2002.10032 (2020)
4. Ballé, J., Laparra, V., Simoncelli, E.P.: End-to-end optimized image compression. arXiv preprint arXiv:1611.01704 (2016)
5. Ballé, J., Minnen, D., Singh, S., Hwang, S.J., Johnston, N.: Variational image compression with a scale hyperprior. arXiv preprint arXiv:1802.01436 (2018)
6. Bellard, F.: Bpg image format (http://bellard.org/bpg/). Accessed: 30 Jan 2021
7. Chen, Y., et al.: Drop an octave: reducing spatial redundancy in convolutional neural networks with octave convolution. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3435–3444 (2019)
8. Cheng, Z., Sun, H., Takeuchi, M., Katto, J.: Learned image compression with discretized gaussian mixture likelihoods and attention modules. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,. pp. 7939–7948 (2020)
9. Covell, M., et al.: Target-quality image compression with recurrent, convolutional neural networks. arXiv preprint arXiv:1705.06687 (2017)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
11. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)
12. Hu, Y., Yang, W., Liu, J.: Coarse-to-fine hyper-prior modeling for learned image compression. Proceedings of the AAAI Conference on Artificial Intelligence. **34**, 11013–11020 (2020)
13. Johnston, N., et al.: Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4385–4393 (2018)
14. Kodak, E.: Kodak lossless true color image suite (photocd pcd0992). URL http://r0k.us/graphics/kodak **6** (1993)

15. Lee, J., Cho, S., Beack, S.K.: Context-adaptive entropy model for end-to-end optimized image compression. arXiv preprint arXiv:1809.10452 (2018)
16. Li, M., Zuo, W., Gu, S., Zhao, D., Zhang, D.: Learning convolutional networks for content-weighted image compression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3214–3223 (2018)
17. Lim, B., Son, S., Kim, H., Nah, S., Mu Lee, K.: Enhanced deep residual networks for single image super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 136–144 (2017)
18. Liu, J., Lu, G., Hu, Z., Xu, D.: A unified end-to-end framework for efficient deep image compression. arXiv preprint arXiv:2002.03370 (2020)
19. Mentzer, F., Agustsson, E., Tschannen, M., Timofte, R., Van Gool, L.: Conditional probability models for deep image compression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4394–4402 (2018)
20. Minnen, D., Ballé, J., Toderici, G.: Joint autoregressive and hierarchical priors for learned image compression. arXiv preprint arXiv:1809.02736 (2018)
21. Ohm, J.R., Sullivan, G.J.: Versatile video coding-towards the next generation of video compression. In: Picture Coding Symposium, vol. 2018 (2018)
22. Oord, A.v.d., Kalchbrenner, N., Vinyals, O., Espeholt, L., Graves, A., Kavukcuoglu, K.: Conditional image generation with pixelcnn decoders. arXiv preprint arXiv:1606.05328 (2016)
23. Rabbani, M., Joshi, R.: An overview of the jpeg 2000 still image compression standard. Sig. Process. Image Commun. **17**(1), 3–48 (2002)
24. Rippel, O., Bourdev, L.: Real-time adaptive image compression. In: International Conference on Machine Learning. pp. 2922–2930. PMLR (2017)
25. Santurkar, S., Budden, D., Shavit, N.: Generative compression. In: 2018 Picture Coding Symposium (PCS), pp. 258–262. IEEE (2018)
26. Sullivan, G.J., Ohm, J.R., Han, W.J., Wiegand, T.: Overview of the high efficiency video coding (HEVC) standard. IEEE Trans. Circuits Syst. Video Technol. **22**(12), 1649–1668 (2012)
27. Theis, L., Shi, W., Cunningham, A., Huszár, F.: Lossy image compression with compressive autoencoders. arXiv preprint arXiv:1703.00395 (2017)
28. Toderici, G., et al.: Variable rate image compression with recurrent neural networks. arXiv preprint arXiv:1511.06085 (2015)
29. Toderici, G., et al.: Full resolution image compression with recurrent neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5306–5314 (2017)
30. Wallace, G.K.: The jpeg still picture compression standard. IEEE Trans. Consum. Electr. **38**(1), xviii-xxxiv (1992)