HDR Image Compression with Convolutional Autoencoder

Fei Han*, Jin Wang*, Ruiqin Xiong†, Qing Zhu* and Baocai Yin*
*Faculty of Information Technology, Beijing University of Technology, Beijing, China
†Institute of Digital Media, Peking University, Beijing, China
Email: feihan@emails.bjut.edu.cn, {ijinwang,ccgszq,ybc}@bjut.edu.cn, rqxiong@pku.edu.cn

***Abstract*—As one of the next-generation multimedia technology, high dynamic range (HDR) imaging technology has been widely applied. Due to its wider color range, HDR image brings greater compression and storage burden compared with traditional LDR image. To solve this problem, in this paper, a two-layer HDR image compression framework based on convolutional neural networks is proposed. The framework is composed of a base layer which provides backward compatibility with the standard JPEG, and an extension layer based on a convolutional variational autoencoder neural networks and a post-processing module. The autoencoder mainly includes a nonlinear transform encoder, a binarized quantizer and a nonlinear transform decoder. Compared with traditional codecs, the proposed CNN autoencoder is more flexible and can retain more image semantic information, which will improve the quality of decoded HDR image. Moreover, to reduce the compression artifacts and noise of reconstructed HDR image, a post-processing method based on group convolutional neural networks is designed. Experimental results show that our method outperforms JPEG XT profile A, B, C and other methods in terms of HDR-VDP-2 evaluation metric. Meanwhile, our scheme also provides backward compatibility with the standard JPEG.**

I. Introduction

In recent years, HDR technology has been greatly developed and applied in areas such as video, games and photograph, for its wider color range makes HDR images closer to the human visual system (HVS). However, the most challenging problem of HDR technology is how to display and store HDR images. The most ordinary devices can only display LDR images with a bit depth of 8 bits for each channel (LDR images pixel values range between 0 and 255). In order to solve the display problem of HDR images, the tone-mapping operators (TMO) have been proposed and used to convert HDR images to LDR images [1]. Compared with LDR images, HDR images bring greater burden to storage and transmission due to its higher brightness range. At the present time, the LDR images are still the most common in digital images. For most users, if their equipment can only decode and display LDR images, and cannot display HDR images, they will know nothing about HDR images information. Consequently, the method of HDR image compression should provide compatibility for users who can only display LDR images. In this regard, the JPEG committee has made great progress. JPEG XT is a two-layer HDR image compression standard issued by the JPEG committee [2, 3], and it encodes the information of the HDR image into the base layer codestream and the extension layer codestream. The base layer is used to provide backward compatibility with the traditional JPEG standard (ISO / IEC 10918), and the extension layer facilitates the compression reconstruction of HDR image. Since then, the two-layer HDR image compression structure has been generally recognized and used.

HDR image compression has achieved satisfactory results using a two-layer structure [2, 3, 5, 13]. There are also a lot of work worth mentioning in LDR image compression, such as encouraging results have been achieved using autoencoder neural networks for image compression [6, 7, 9, 10]. Our framework is a two-layer framework based on convolutional neural networks, which mainly includes a base layer, a new extension layer and a post-processing module. The base layer is the traditional base layer and it is used to provide backward compatibility with the traditional JPEG standard. The extension layer consists of a convolutional variational autoencoder neural networks. Our autoencoder is a process of non-linear dimensionality reduction and dimensionality enhancement of residual image features. The semantic information retained in this process is conducive to the generation of extension layer codestreams, and it improves the efficiency of compression and reconstruction. The autoencoder mainly includes a residual encoder E, a binarizer B and a residual decoder D. The post-processing module is based on a group convolutional neural networks, and the post-processing module is proposed to improve the quality of original HDR image reconstrucion.

II. Proposed Method

In this paper, we propose a two-layer HDR image compression framework based on convolutional neural networks that is backward compatible with the JPEG standard. The details of our framework are shown in Fig. 1.

Fig. 1(a) shows the HDR encoder. First, the original HDR image is tone-mapped to obtain the LDR image. A logarithmic function is applied to perform TMO to facilitate the implementation. Then, a standard JPEG encoder and decoder (quality q) are used on the LDR image to generate the base layer codestream and reconstructed LDR image. Next, all HDR values that have passed through TMO to the same LDR value are averaged, and a lookup table is created for the inverse tone-mapping operators(ITMO). The reconstructed LDR image will generate a new HDR image after ITMO. Finally, the residual value is the difference between the original HDR image and the new HDR image. The residual value is normalized and mapped between 0 and 255:

$$res = \frac{Res - Min}{Max - Min} \tag{1}$$

where Min, Max are the minimum and maximum values in the residual image. At the extension layer, a CNN autoencoder is used to encode the residual image. Our residual encoder extracts the high level features of the image, which will be

978-1-7281-8068-7/20/$31.00 ©2020 IEEE

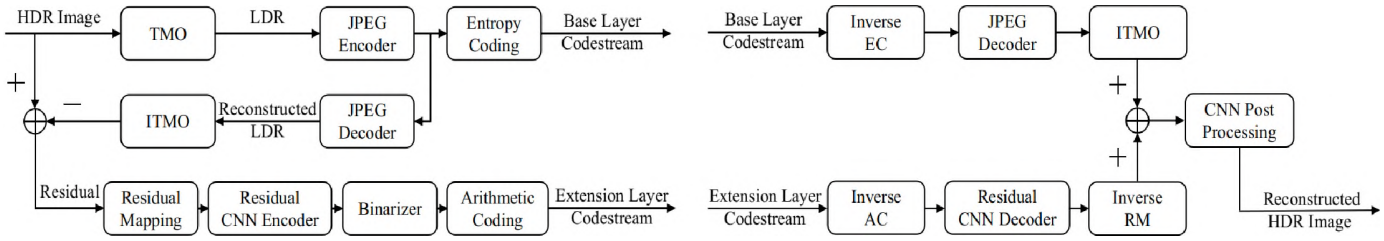

Fig. 1. The proposed framework of backward compatible two-layer HDR image compression.

more conducive to the representation of the image and reduce the length of feature codestream. Then binary quantization and arithmetic coding are performed to generate the extension layer codestream.

The HDR decoder is as shown in Fig. 1(b). Inverse EC and inverse AC represent inverse entropy coding and inverse arithmetic coding, respectively. First, the base layer codestream is decoded into a reconstructed LDR image by a standard JPEG decoder, which provides compatibility with the traditional JPEG standard. The reconstructed LDR image passes through ITMO and then generates a new HDR image. Then, the extension layer codestream is decoded by the decoder in the CNN autoencoder. Our residual decoder contains more residual semantic information, which will help to improve the reconstruction quality and compression performance of the residual. After that the inverse mapping is used to generate the reconstruction residual. Finally, the reconstructed residual and the new HDR image are added together to form HDR image, and then the final reconstructed HDR image is generated after a post-processing module based on a group convolutional neural networks.

A. Residual Encoder And Decoder

Our residual encoder and decoder architecture are shown in Fig. 2. The residual encoder and decoder are based on CNN autoencoder. Residual decoder and encoder are composed of convolution, deconvolution and GDN/IGDN nonlinear activation function. The GDN/IGDN activation function is a local division normalization conversion, which has proved to be particularly suitable for evaluating image quality for image compression [5, 6]. In the residual encoder, we design a feature fusion structure to concatenate and fuse features of different convolutional layers, by using a concatenate operation is used to fuse features of different convolutional layers. The original residual image is down-sampled to $1/2$, $1/4$, $1/8$, $1/16$ by the convolutional layer. Then the down-sampled features is applied with a 1×1 convolution to generate the encoded representation y. In the end of encoder, the original residual image ($H \times W \times 3$) is down-sampled to a feature map of $H/16 \times W/16 \times 32$ size. In the residual decoder, we up-sample the coded stream $\hat{y}$ to a reconstruction residual image of $H \times W \times 3$ size through the intermediate deconvolution layer and convolution layer.

B. Binarizer

In the proposed framework, a binarizer is used in our quantization. Compared with traditional quantization, the binarizer

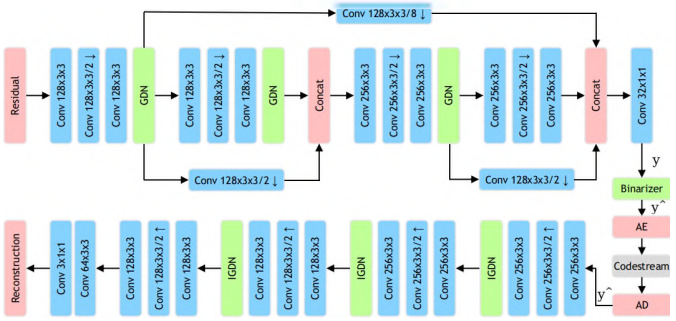

Fig. 2. Illustration of the variational autoencoder neural networks used in residual encoder and decoder. Convolution parameters are denoted as number of filter × kernel height × kernel width/ down or upsampling stride, where indicates down-sampling (Convolution) andindicates up-sampling (Deconvolution). AE and AD represent arithmetic encoder and decoder. Codestream stands for the extension layer codestream.

can further control and reduce the length of code stream, which will help CNN autoencoder to learn effective feature representation. Moreover, simple calculation of binarizer can reduce training time. The binarization process consists of two parts. In the first step, the sigmoid activation function is used after the last 1×1 convolutional layer of the residual encoder, the residual encoder output $y = E(x)$ should be in the range of $[0, 1]$. For the second part, let y_{ijk} denote the elements in y. The binarizer $B(E(x))$ is defined as follows:

$$B(y_{ijk}) = \begin{cases} 0, & if\ y_{ijk} \leq 0.5 \\ 1, & if\ y_{ijk} > 0.5 \end{cases} \tag{2}$$

However, the gradient of the $B(y_{ijk})$ function is 0 almost everywhere except that it is infinite when $y_{ijk} = 0.5$. In the back-propagation algorithm, this will result in the inability to update and reduce the gradient. Recently, some helpful works on binarizer have studied the problem of gradient update in back propagation through binarization [8, 10, 11]. Inspired by these works, a generalization function $\tilde{B}(y_{ijk})$ is designed to approximate $B(y_{ijk})$ function as follows:

$$\tilde{B}(y_{ijk}) = \begin{cases} y_{ijk} + \epsilon, & if\ 0 \leq y_{ijk} \leq 1 \\ 0\ or\ 1, & otherwise \end{cases} \tag{3}$$

where ϵ is random noise. The $B(y_{ijk})$ function is only used in forward propagation calculation, while $\tilde{B}(y_{ijk})$ is used in back propagation. The gradient of $\tilde{B}(y_{ijk})$ function can be obtained

by the following formula:

$$\tilde{B}'(y_{ijk}) = \begin{cases} 1, & if\ 0 \le y_{ijk} \le 1 \\ 0, & otherwise \end{cases} \quad (4)$$

C. Residual Reconstruction And Rate Control

As shown in Fig. 3, inspired by [8, 9], we use an iterative accumulation way to control the extension layer bit rate and reconstruct the original residual image. In the first iteration, the input image of the encoder is the original residual image. In each subsequent iteration, the input image is the residual, the output image is the prediction of the residual, and the residual here refers to the difference between the input image and the output image of the previous iteration. The final residual image reconstruction is then the sum of the output images in all iterations. In our autoencoder, each $256\times256\times3$ input residual image is reduced to a $16\times16\times32$ binarized representation per iteration. If the representation of each binarized feature value consumes 1 bit, the result in each iteration represents $1/8$ bit per pixel (bpp). Even before using entropy encoding, the first iteration can achieve $192:1$ compression ratio. As the number of iterations increasing, the bit rate increases 0.125 bpp each time, and the final bit rate will reach 2 bpp after 16 iterations.

D. Post-processing Based on CNN

After the initial reconstruction is obtained, there may be compression artifacts and over-smooth texture details. In order to improve the reconstruction quality of HDR images, an effective post-processing module is designed as shown in Fig. 4. Our post-processing module is a two-layer group convolutional neural networks. The network is mainly composed of 3×3 and 5×5 residual blocks. This block has been widely used in image processing, such as image compression [10] and denoising [12]. We use 10 residual blocks in each group of neural networks, and such a deep networks can further improve the quality of image reconstruction.

III. Experimental Results

A. Experimental Settings

In the neural network training of the residual autoencoder, we use 256×256 residual image blocks as input images. The original HDR image comes from the public Internet HDR image dataset and video sequence[15] (including HDReye, Fairchild, Funt, MPI, etc.). First, we decompose these HDR images into non-overlapping 256×256 image blocks, and then perform data enhancement on these HDR image blocks. Data enhancement includes flipping, rotating and setting different quality q in the traditional JPEG codec. In the end, we get about 0.35 million residual image blocks. As the large difference between the maximum and minimum values of the residual image values, in order to facilitate residual training and image reconstruction, we map the residual image values to $[0, 255]$. No inverse residual mapping is used in the training phase, and inverse residual mapping is only used in the test phase. The new HDR image after the addition operation will be the input image of the post-processing. Mean square error(MSE) is used as distortion metric, and the distortion loss function is as follows:

$$L = ||x - \hat{x}||^2 \quad (5)$$

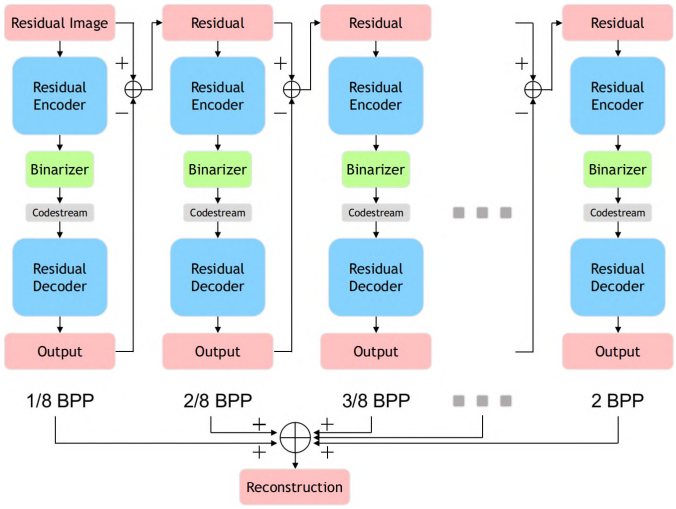

Fig. 3. Illustration of the residual reconstruction method based on iteration and accumulation.

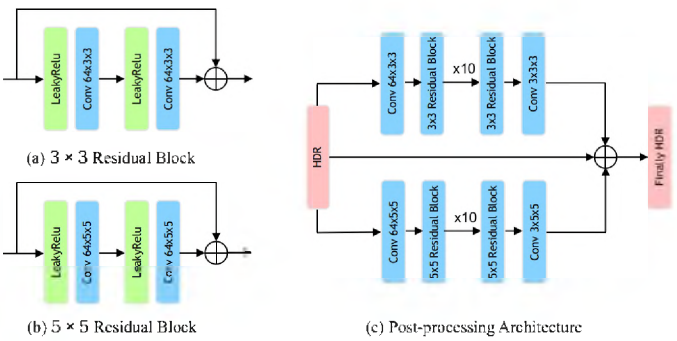

Fig. 4. (a) The 3×3 residual block. (b) The 5×5 residual block. (c) The post-processing architecture is composed of two groups of convolutional networks, and each group contains 10 residual blocks and two convolutional layers.

where x is input image, $\hat{x}$ is output image. We are training on Linux operating system and Pytorch framework, with Tesla V100 GPU.

B. Results And Analysis

In the HDR image quality evaluation, the HDR-VDP-2 index is adopted as the objective evaluation metric, which is a widely used metric to estimate the quality of HDR images [14]. The test images are from Ward's HDR image dataset [15], and we compare our method with JPEG XT profile A, B, C and Li[13], where JPEG XT reference implementation comes from [15].

As shown in Fig. 5, in the objective image quality evaluation, our method is superior to JPEG XT profile A, B, C and Li's method at low and medium bit rates, and the performance at high bit rate is not as good as that of Li's method. In our method, the gradient of the rate-distortion curve increases rapidly at low and medium bit rates, and the gradient at high bit rates tends to be stable. The reason is that we use iterative accumulation to reconstruct the image, and in the previous iterations, the effect of image reconstruction quality improvement after each iteration is obvious. However, as the number of iterations increasing, the reduction in the residual between the original image and the reconstructed

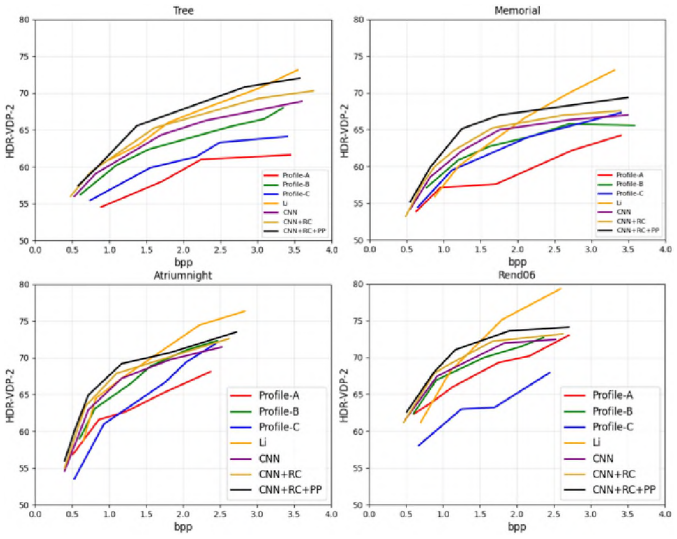

Fig. 5. Objective image quality comparison with JPEG XT and Li. (In our method, CNN stands for autoender, RC represent arithmetic coding, and PP stands for post-processing module.)

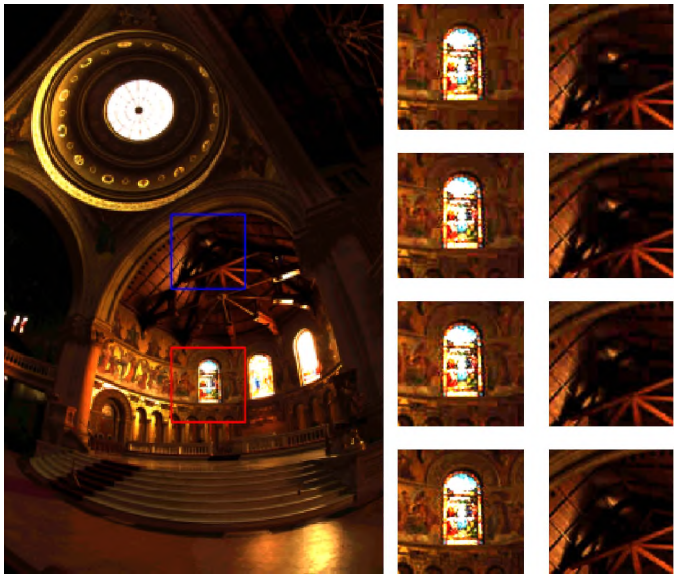

Fig. 6. Subjective image quality comparison with JPEG XT. (JPEG XT profile A, B, C and our methods (CNN+RC+PP) from top to bottom, their bit rate is fixed at 3.26bpp, 3.19bpp, 3.23bpp, 3.05bpp, respectively.)

image becomes smaller and smaller, the residual consumes the same bit rate per iteration (0.125bpp), but the magnitude of the increase in image reconstruction quality is decreasing. In terms of ablation experiments, we perform ablation experiments with autoencoder alone(CNN), autoencoder and arithmetic coding(RC), and autoencoder together with arithmetic coding and post-processing module(PP), the results are also as shown in Fig.5. The experiments showed that RC can achieve gain of about 0.5 to 1.0 db on HDR-VDP-2, and PP can be improved by about 0.5 to 1.5 db. At the same time, we also make a comparative assessment of subjective image quality (as show in Fig. 6), the red box is the bright area, and the blue box is the dark area. We can see that the subjective image quality of our method is slightly better than JPEG XT profile A, B, C.

IV. Conclusion

In this paper, a backward compatible HDR image compression framework based on convolutional neural networks is proposed. The framework is divided into base layer and extension layer. The base layer can provide backward compatibility with the traditional JPEG standard. The residual layer is encoded and decoded by the convolutional autoencoder, and finally the post-processing module improves the image reconstruction quality. In the HDR image quality evaluation metric (HDR-VDP-2), our method has the best performance at low and medium bit rates compared to the JPEG XT profile A, B, C and Li[13]. In future work, we hope to explore more effective HDR image compression autoencoder by optimizing the network architecture.

Acknowledgment

This work is supported by NSFC(No.61906008, 61672066, 61976011), the Beijing Key Laboratory of Multimedia and Intelligent Software Technology, Beijing Artificial Intelligence Institute, Scientific Research Project of Beijing Educational Committee(KM202010005013), and the Opening Project of Beijing Key Laboratory of Internet Culture and Digital Dissemination Research.

References

[1] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, "Photographic tone reproduction for digital images," SIGGRAPH, 21(3), pp. 267-276, 2002.

[2] A. Artusi, R. K. Mantiuk, T. Richter, and P. Korshunov, "Jpeg xt: a compression standard for hdr and wcg images," IEEE Signal Process Mag, 33(2), pp. 118-124, 2016.

[3] A. Artusi, R. K. Mantiuk, T. Richter, P. Hanhart, P. Korshunov, ... and T. Ebrahimi, "Overview and evaluation of the JPEG XT HDR image compression standard," JRTIP, 16(2), pp. 413-428, 2019.

[4] M. Iwahashi, and H. Kiya, "Two layer lossless coding of HDR images," ICASSP, pp. 1340-1344, 2013.

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

[6] 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.

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

[8] G. Toderici, S. M. O'Malley, S. J. Hwang, D. Vincent, D. Minnen, S. Baluja, ... and R. Sukthankar, "Variable rate image compression with recurrent neural networks," arXiv preprint arXiv:1511.06085, 2015.

[9] G. Toderici, D. Vincent, N. Johnston, S. J. Hwang, D. Minnen, J. Shor, and M. Covell, "Full resolution image compression with recurrent neural networks," CVPR, pp. 5306-5314, 2017.

[10] L. Zhou, C. Cai, Y. Gao, S. Su, and J. Wu, "Variational Autoencoder for Low Bit-rate Image Compression," CVPR, pp. 2617-2620, 2018.

[11] M. Li, W. Zuo, S. Gu, D. Zhao, and D. Zhang, "Learning convolutional networks for content-weighted image compression," CVPR, pp. 3214-3223, 2018.

[12] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," IEEE TIP, 26(7), pp. 3142-3155, 2017.

[13] S. D. Li, J. Wang, Q. Zhu, "High Dynamic Range Image Compression based on Visual Saliency," PCS, pp. 21-25, 2018.

[14] R. Mantiuk, K. J. Kim , A. G. Rempel, and W. Heidrich, "Hdr-vdp-2: a calibrated visual metric for visibility and quality predictions in all luminance conditions," ACM TOG, 30(4), pp. 40.1-40.13, 2011.

[15] https://github.com/HFJJ/HDR_DataSet. This is a link which include training dataset, test dataset, and JPEG XT reference implementation.