Learned Prior Information for Image Compression

Ching-Chun Huang Thanh-Phat Nguyen Chen-Tung Lai
Dept. of Electrical Engineering and Advanced Institute of Manufacturing with High-tech Innovations,
National Chung Cheng University, Chiayi, Taiwan
Center for Innovative Research on Aging Society (CIRAS)
{chingchun.huang6,nguyenthanhphat2112,lai.chen.tung1219}@gmail.com

Abstract

We propose a method for image compression by integrating a deep neural network (DNN) with the better portable graphics (BPG) codec. As DNN can learn the prior information from image data, it will reduce the transmission information through BPG codec and achieves a good visual quality for the decompressed image. The proposed method includes three parts: the BPG codec, the artifact reduction network and the colorization network. First, image is converted to the CIE Lab color space. Then, the BPG codec compresses L component and color hint extracted from the a, b components. To satisfy the file size, the suitable QP values of BPG compression will be found for each image by binary search. Next, the decompressed L will be improved by the artifact reduction network. Finally, the colorization network will predict a and b components from the decompressed L and the color hint. We evaluate the proposed method upon the CLIC validation sets and Kodak image sets by the quantitative metrics (PSNR, MS-SSIM). The comparison with BPG is also presented.

1. Introduction

Lossy image compression, which aims to encode the image using fewer bits, has been extensively researched for decades. The traditional image coding, e.g., JPEG [1] and JPEG 2000 [2], are generally designed based on the fixed hand-crafted image transformation such as Discrete Cosine Transform (DCT) or Discrete Wavelet Transform (DWT). Recently, the better portable graphics (BPG) codec [3], which leverages intra-frame coding tools in the high-efficiency video coding (HEVC) for image compression has outperformed other standard lossy image coding methods.

In recent years, neural networks and deep learning have led to significant achievements in various computer vision tasks. With the advantage of automatic feature extraction and high-level representation, many works have applied end-to-end deep learning to image compression with different autoencoder architectures and quantization methods such as [4–6]. However, the end-to-end image compression network has not been ready to beat the state of the art image compression codec such as BPG codec. Instead of end-to-end learning for image compression, some works were performed as a post-processing method for the standard image coding such as compression artifact reduction [7–9]. These methods can improve quality of the decompressed image, but they may not reduce the transmit information through compressor. One possible solution to perform this reduction is colorization method [10], which only requires the grayscale image and few guidance information. If the colorization method is good enough, the compressor only needs to compress the graycale image and guidance information, which ocupies lower bits than the color image.

Hence, in this work, we proposed a method for image compression that includes three parts: BPG codec, which is used as compressor and decompressor; artifact reduction network, which improves the decompressed image; and colorization network, which leverages prior information to predict the color component of image.

This work was partially supported by Ministry of Science and Technology of Taiwan under Grant No. 108-2634-F-009 -006, 108-2634-F-009 -007, 107-2218-E-009 -062, and 106-2628-E-194 -002 -MY3.

2. Related Works

Deep learning based methods for compression artifact reduction consist Dong et al. [7], which proposes 4 layer convolutional neural network with specific operations: feature extraction, feature enhancement, mapping and reconstruction for deblocking and deblurring of compressed images; Zheng et al. [9], which proposes an greedy loss architecture for improving the JPEG compression artifact reduction; and Kirmemis et al. [8], which proposes a solution based on residual blocks to reduce compression artifacts for BPG compressed image.

On the other hand, the deep learning based colorization networks include Endo et al. [11], which automatically learns similarity between pixels given user strokes and in-

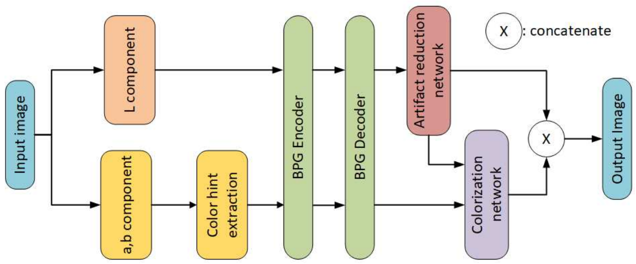


Figure 1: The proposed method architecture

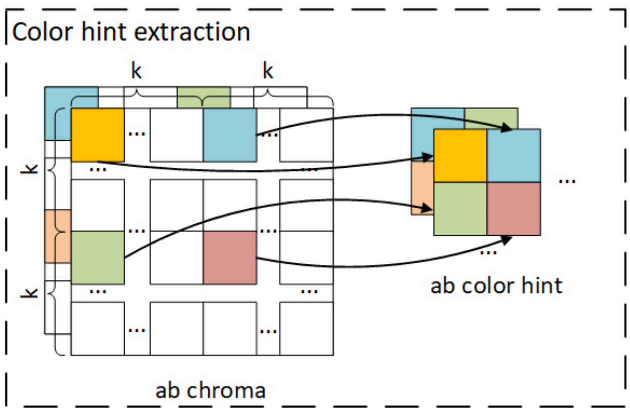


Figure 2: Color hint extraction

put images; Sangkloy et al. [12], which developed a system to translate sketches to real images, with support for user color strokes; Zhang et al. [10], which proposed a U-Net structure to generate color image from the grayscale image and the color hint.

3. Proposed method

The proposed framework for image compression is shown in Figure 1. The whole framework is divided into three main components: the compression module based on BPG codec, the artifact reduction network, and the colorization network.

3.1. Compression module based on BPG Codec

First, the original image is converted to the CIE Lab color space. Then, the color hint is generated by getting the top left corner value of every k $\times$ k pixels in the chroma a and b, as shown in Figure 2. Finally, L component and color hint will be compressed by the BPG Codec with the corresponding target bitrate 0.11 and 0.04. In order to satisfy the bitrate condition, the suitable QP value of BPG encoder will be found by using binary search. Hence, the range of QP value is set in the range [1, 51] and the binary search will iteratively find the QP value inside this range until the real bitrate aproximates to the target bitrate.

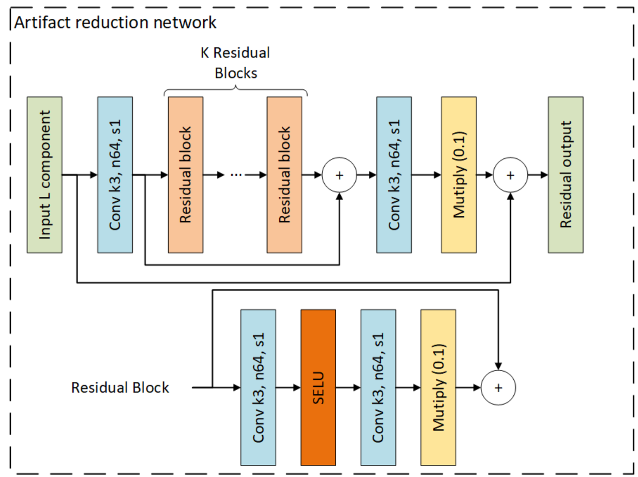


Figure 3: Artifact Reduction Network

3.2. Artifact Reduction Network

During BPG compression, the color hint will have a bitrate as 0.04$\times$k$\times$k bpp. When k is bigger, bit-rate increases exponentially, which leads to less artifact in the compression result without the need of post-processing. In contrast, the decompressed L may contain some unexpected artifacts, which reduces quality of the recovered color image. To overcome this problem, the proposed artifact reduction network [8] will be leveraged. Instead of directly generating an image, the network will be modified to predict the residual output, which later is added to the input L component for achieving better L component. The structure of network is illustrated in Figure 3. It is composed of 6 layers including 2 convolutional layers, 4 residual blocks. The input is the L image after compression and the output is the improved L. The loss function is defined as Eqn. 1.

$$L_a = \frac{1}{N}\sum_{i=1}^{N}\sum_{h,w}||R_i - (\hat{L}_i - L_i)||_2^2 \qquad (1)$$

where R_i is the residual output of the network; $\hat{L}_i$ is the ground truth L component; L_i is the input decompressed L component; N is the number of training images; h and w are the index of height and width of an input image, respectively.

3.3. Colorization Network

Before this step, the decompressed color hint needs to be reshaped to its original size by performing inverse process in the compression step and all empty pixels are set to zero. Then, the colorization network with local hint condition [10] is used to generate the color component a and b. The inputs to network are a grayscale image $X \in \mathbb{R}^{H\times W\times 1}$, along with the color hint $U \in \mathbb{R}^{H\times W\times 2}$ and the binary mask $B \in \mathbb{R}^{H\times W\times 1}$ indicating which pixels contain the hint. The output of the network is $Y \in \mathbb{R}^{H\times W\times 2}$, the estimation of

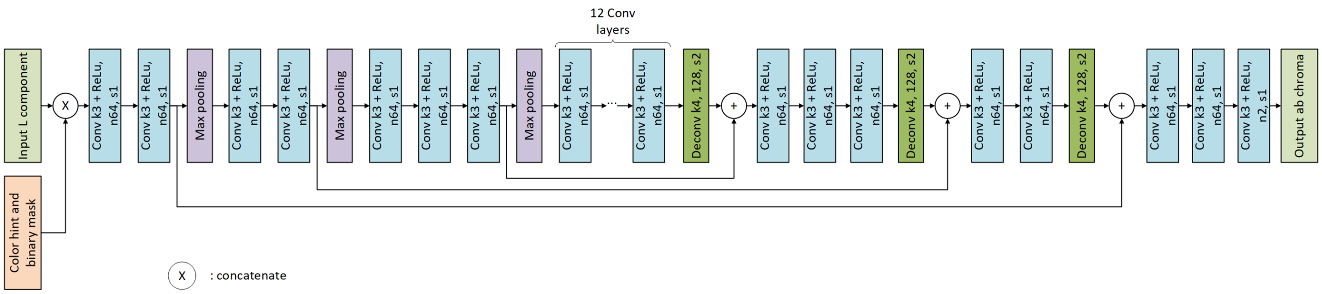


Figure 4: Colorization Network

the a and b color channels of the image. The structure of the colorization network is shown in Fig. 4. The network is trained to minimize the objective function in Eqn. 2.

$$L_c = \sum_{j=1}^{M}\sum_{h,w}\left[\left(\frac{1}{2}(Y^j_{h,w} - \hat{Y}^j_{h,w})^2 1_{\{|Y^j_{h,w}-\hat{Y}^j_{h,w}|<\delta\}}\right) + \delta\left(|Y^j_{h,w} - \hat{Y}^j_{h,w}| - \frac{1}{2}\delta) 1_{\{|Y^j_{h,w}-\hat{Y}^j_{h,w}|>\delta\}}\right)\right] \quad (2)$$

where Y is the predicted a, b from the network; $\hat{Y}$ is the ground truth of a, b components; M is the number of training images; h, w are the index of height and width of an output image, respectively; and δ is the threshold, in this work $\delta = 1$.

In order to obey RAM requirements (16 GB) on the evaluation server, the input L and color hint are divided into 4 blocks and the colorization network processes these blocks separately. Output blocks of the colorization network are merged together to get the final ab chroma.

4. Experimental Results

4.1. Training setting

We train the artifact reduction network and the colorization network separately. The artifact reduction network is trained with mobile training set consisting of 1048 images and batch size of 64, while the colorization network is trained with Imagenet dataset ILSVRC2012 [13] and batch size of 25. For the artifact reduction network, we use BPG encoder to encode the training L images with the target bitrate around 0.11. We train both networks using the Adam optimizer [14] with the default parameters (β_1 = 0.9, β_2=0.999) The learning rate is initialized to 0.0001 and is halved at every 100^{th} epoch. Networks are trained on 256x256 random crops without any data augmentation. We stop training networks when the learning curve converges.

Table 1: Average PSNR and MS-SSIM for the validation set

	Validation Set		
Method	PSNR	MS-SSIM	bpp
BPG	**37.27**	**0.9468**	0.1487
Ours	34.2293	0.9379	**0.1467**

Table 2: Average PSNR and MS-SSIM for the Kodak Dataset

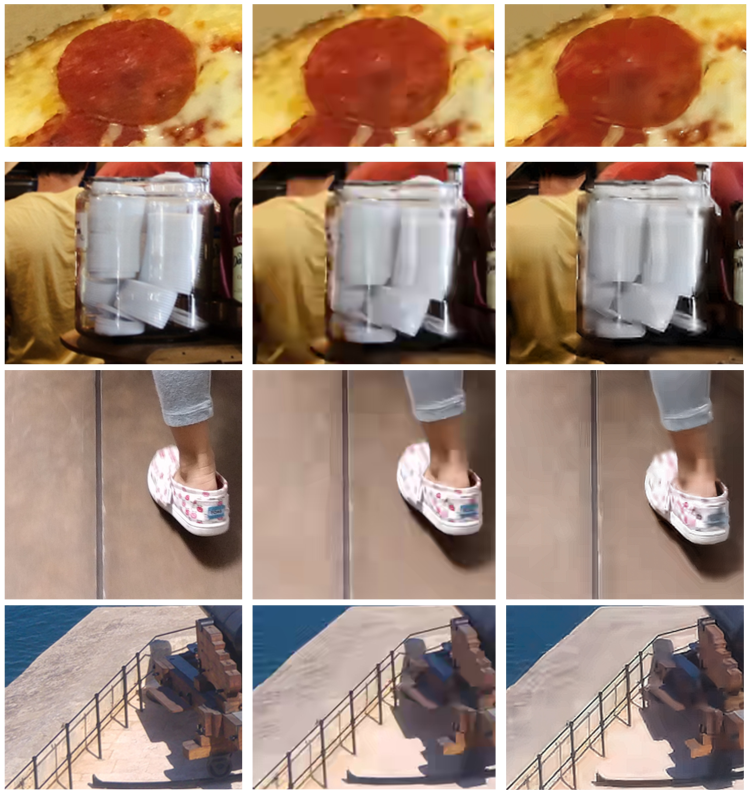
	Kodak Dataset		
Method	PSNR	MS-SSIM	bpp
BPG	**28.8446**	**0.9206**	**0.1465**
Ours	28.2318	0.9150	0.1476

4.2. Evaluation

We present PSNR and MS-SSIM results for the given validation set consisting of 102 images. Table 1 shows the average PSNR and MS-SSIM results of our method and BPG on the validation sets. The proposed method performs with PSNR lower than BPG and MS-SSIM similar to BPG. We also evaluate methods on the Kodak dataset [15], which consists of 24 images. Evaluation results are shown in Table 2. Although our results can not exceed the BPG method, the visual results of our method are better than the BPG result in some cases, as shown in Figure 5.

5. Conclusions

This paper describes an image compression approach which combine BPG codec and Deep Learning methods. Our results show that the average PSNR and MS-SSIM are close to the BPG result. By combining the deep neural network, we can satisfy the file size requirement and the compression time. Although the proposed method can not outperform BPG, it shows that a combination of deep neural network and the standard image coding will be a potential



Left: Original cropped images. Middle: BPG compression results. Right: The proposed method results

Figure 5: Visual results for images in the Validation Set

direction to perform image compression.

References

[1] G. K. Wallace. The jpeg still picture compression standard. *Communications of the ACM*, pages 30–44, 1991.

[2] Majid Rabbani and Rajan Joshi. An overview of the jpeg2000 still image compression standard. In *Signal Processing: Image Communication*, pages 3–48, 2002.

[3] Better portable graphics encoder/decoder and bitstream specification. https://bellard.org/bpg/, https://bellard.org/bpg/bpg_spec.txt. Accessed: 2019-05-05.

[4] J. Ballé, V. Laparra, and E. P. Simoncelli. End-to-end optimized image compression. In *ICLR*, 2017.

[5] L. Theis, W. Shi, A. Cunningham, and F. Huszár. Lossy image compression with compressive autoencoders. In *ICLR*, 2017.

[6] M. Li, W. Zuo, S. Gu, D. Zhao, and D. Zhang. Learning convolutional networks for content-weighted image compression. In *CVPR*, pages 3214–3223, 2018.

[7] C. Dong, Y. Deng, L. C. Change, and X. Tang. Compression artifacts reduction by a deep convolutional network. In *ICCV*, 2015.

[8] O. Kirmemis, G. Bakar, and A. Murat Tekalp. Learned compression artifact removal by deep residual networks. In *CVPR Workshops*, 2018.

[9] B. Zheng, R. Sun, X. Tian, and Y. Chen. S-net: A scalable convolutional neural network for JPEG compression artifact reduction. *CoRR*, abs/1810.07960, 2018.

[10] R. Zhang, J. Y. Zhu, P. Isola, X. Geng, A. S Lin, T. Yu, and A. A Efros. Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics (TOG)*, 9(4), 2017.

[11] Y. Endo, S. Iizuka, Y. Kanamori, and J. Mitani. Deepprop: Extracting deep features from a single image for edit propagation. *Comput. Graph. Forum*, 35(2):189–201, 2016.

[12] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *CVPR*, pages 6836–6845, 2017.

[13] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015.

[14] Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[15] Kodak photocd dataset. http://r0k.us/graphics/kodak/. Accessed: 2019-05-05.