DEEP IMAGE COMPRESSION WITH ITERATIVE NON-UNIFORM QUANTIZATION

*Jianrui Cai and Lei Zhang**

Dept. of Computing, The Hong Kong Polytechnic University, Hong Kong
Email:{csjcai, cslzhang}@comp.polyu.edu.hk

ABSTRACT

Image compression, which aims to represent an image with less storage space, is a classical problem in image processing. Recently, by training an *encoder-quantizer-decoder* network, deep convolutional neural networks (CNNs) have achieved promising results in image compression. As a non-differentiable part of the compression system, quantizer is hard to be updated during the network training. Most of existing deep image compression methods adopt a uniform rounding function as the quantizer, which however restricts the capability and flexibility of CNNs in compressing complex image structures. In this paper, we present an iterative non-uniform quantization scheme for deep image compression. More specifically, we alternatively optimize the *quantizer* and *encoder-decoder*. When the encoder-decoder is fixed, a non-uniform quantizer is optimized based on the distribution of representation features. The encoder-decoder network is then updated by fixing the quantizer. Extensive experiments demonstrate the superior PSNR index of the proposed method to existing deep compressors and JPEG2000.

***Index Terms*—** Deep Image Compression, Iterative Non-Uniform Quantization

1. INTRODUCTION

With the explosive growth of portable imaging devices and social media (*e.g.*, Facebook and Flickr), billions of images are shared daily on social networks. Image compression, especially lossy image compression, is a must to reduce the storage space and provide an economic solution to a wide range of image storage and transmission systems.

A typical lossy image compression system consists of an encoder, a quantizer and a decoder. Popular lossy image compression systems (*e.g.*, JPEG and JPEG2000) are generally designed in a hand-crafted manner. Pre-defined transformations (*e.g.*, DCT for JPEG and DWT for JPEG2000) are firstly applied to transform the images into a sparse domain, where quantization is then performed on the transformed coefficients, followed by entropy coding. Though simple and efficient to implement, the analytically designed DCT and DWT transforms are limited to represent the various complex structures in natural images. As a result, these compression systems may generate severe visual artifacts (*e.g.*, blocky artifacts and ringings) in the decompressed images.

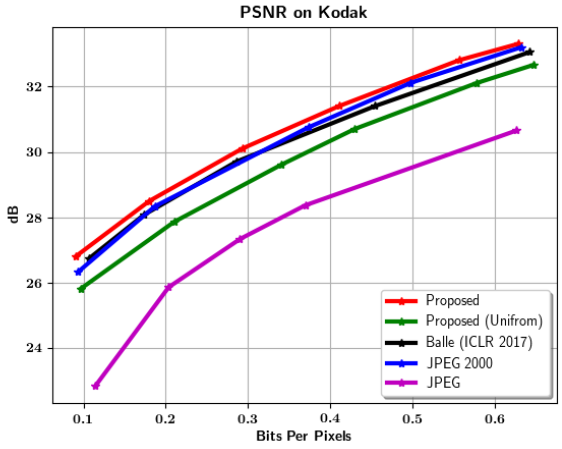

Fig. 1. Comparison of the rate-distortion curves on Kodak.

Deep convolutional neural networks (CNNs) have recently led to a series of breakthroughs in versatile vision problems [1, 2, 3, 4]. The flexible non-linear modelling capability and powerful end-to-end training paradigm of CNN make it a promising new approach for image compression. Firstly, the end-to-end training manner enables CNN-based compression systems to adaptively learn an effective encoder-decoder from a large amount of image data and in a larger context to represent more complex image structures, reducing the artifacts in the decompressed image. Secondly, by adopting specific loss functions (*i.e.*, perceptual metrics) in the training, the CNN compressors are able to strengthen certain desired aspects (*i.e.*, visual quality) of the output image. In the last several years, a flurry of CNN-based image compression works have been proposed, including the study of network structures [5, 6, 7, 8, 9] as well as loss functions [10, 11, 12, 13].

Despite the advantages of employing CNN for compression, there are still some challenges which limit the performance of CNN-based compressor. Quantization, a key component of compression pipeline, is to generate discrete sym-

This work is supported by Hong Kong RGC GRF grant (PolyU 152124/15E).

bols to encode the latent image representation using a finite number of bits. Due to the non-differentiable property of discrete operation, however, quantizer is hard to be updated during the end-to-end CNN network training. Therefore, an optimal quantizer is almost unreachable for the existing deep compression methods.

To address the above mentioned problem, in this paper we propose an iterative non-uniform quantization strategy to train a deep CNN compressor. The *quantizer* and *encoder-decoder* are trained in an alternative optimization manner. With fixed quantizer, an encoder-decoder network is trained to minimize the ℓ_1 loss between the input and reconstructed images. While with the fixed encoder-decoder network, an optimal non-uniform quantizer is adaptively learned based on the distribution of encoding coefficients to minimize the quantization error. The *quantizer* and *encoder-decoder* are alternatively and iteratively updated till convergence. Experimental results validate that the proposed deep image compression algorithm produces better reconstruction results than previous methods in terms of PSNR, as shown in Figure 1.

2. RELATED WORK

Recently, deep neural networks have been investigated and achieved promising results in image compression. As pioneering works, Toderici *et al*. adopted a recurrent neural network (RNN) to encode and decode images of size 32x32 [5], and further extended the network to handle full-resolution images [6]. Built on the architecture proposed in [5, 6], Johnston *et al*. [12] replaced the mean-squared error (MSE) loss with SSIM weighted loss to improve the visual quality of the reconstruction image.

Different from RNN, methods in [7, 8, 9, 10, 13, 11] rely on CNN based auto-encoder architectures. Ballé et al. [7] used generalized divisive normalization (GDN) for joint nonlinearity to implement local gain control. By exploiting the pyramidal decomposition strategy, Rippel *et al*. [10] aggregated information across different scales. Li *et al*. [9] introduced a content-weighted importance map to guide the bit rate allocation. To minimize the gradient vanish effect caused by non-differentiable quantization operation, Theis *et al*. [8] introduced a smooth approximation of the derivative of the rounding function. A soft-to-hard scheme is adopted in [13] to find assignments to the quantizer.

Because of the non-differentiable operation in quantization, the quantizer is often preset in previous deep compression methods, and fixed during networks training. Therefore, quantization intervals and quantized values are usually not optimized. Different from those methods, we propose a new training strategy, which can iteratively update a non-uniform quantizer to reduce quantization error.

3. METHOD

The architecture of our encoder-quantizer-decoder compression network is shown in Figure 2. Given a set of training

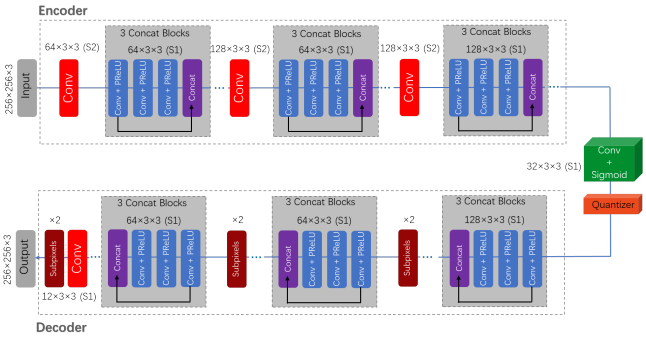

Fig. 2. Illustration of our model architecture. The upper and lower parts represent the encoder **E** and decoder **D**, respectively. The notation $C \times K \times K(SN)$ represents a convolution layer with kernel size K, C output channels and a stride of N. The notation $\times 2$ refers to the upsampling factor.

images X, we aim to learn a nonlinear analysis transformation encoder **E**, a non-uniform quantizer **Q**, and a nonlinear synthesis transformation decoder **D**. The encoder **E** first convert an input image x into a latent representation $z = \mathbf{E}(x)$. Then, a non-uniform quantizer **Q** quantizes the features into discrete intervals with quantized values $\hat{z} = \mathbf{Q}(z)$, which can be losslessly encoded into a bitstream for transmission or storage. Once the bitstream is received by the decoder **D**, an estimate of the original input image is obtained as $\hat{x} = \mathbf{D}(\hat{z})$. Overall, the deep compression framework can be formulated as:

$$\hat{x} = \mathbf{D}(\mathbf{Q}(\mathbf{E}(x, \Omega)), \Theta), \tag{1}$$

where Ω and Θ are the parameters of encoder **E** and decoder **D**, respectively. Given a compression rate, the network is expected to learn the parameters Ω and Θ to minimize the distortion of the reconstructed image.

3.1. Encoder and Decoder

Loss Function: Many existing CNN based deep compressors [10, 11, 13] use the MS-SSIM or adversarial loss to strength the perceptual quality of the compressed image, while ignoring the index of PSNR. We argue that PSNR (or MSE) is still one of the most important factors to evaluate a compressor, and thus aim to optimize the PSNR during the CNN training. We adopt the $\ell 1$-norm of the reconstruction error as the loss function because we experimentally found that a compression network trained with $\ell 1$ loss achieves slightly improved PSNR compared with the $\ell 2$ loss. Such a phenomenon is also reported in [14]. The following loss function is used for our proposed network:

$$l(\Omega, \Theta) = \frac{1}{n} \sum_{i}^{n} \|x_i - \mathbf{D}(\mathbf{Q}(\mathbf{E}(x_i, \Omega)), \Theta)\|_1, \tag{2}$$

where x_i refers to the i-th image in X.

Architecture: Our encoder-decoder network has 4 types of layers which are shown with 4 different colors in Figure 2. Instead of using residual blocks, we adopt concat blocks to concatenate the feature maps of two layers to ensure maximum information flow. The PReLU [15] is adopted as the activation function since it could improve model fitting with nearly zero extra computational cost and little over-fitting risk. Subpixel [16] is adopted to reshape and upsample feature maps.

3.2. Alternative Network and Quantizer Training

Since the quantizer $\mathbf{Q}(\cdot)$ is a discrete function, we cannot differentiate it with respect to its argument, which prevents us from computing the best quantization intervals and quantized values. To address this issue, we first train the encoder-decoder network without the quantizer. The objective function can be rewritten as:

$$l(\Omega, \Theta) = \frac{1}{n} \sum_{i}^{n} \|x_i - \mathbf{D}(\mathbf{E}(x_i, \Omega), \Theta)\|_1. \tag{3}$$

When the parameters Ω and Θ are learned, the latent representation of an image z can be obtained by $z = \mathbf{E}(x, \Omega)$. With a set of latent representations of training images, we can easily compute $p(z)$, the probability density function (PDF) of z. The optimal quantizer can be solved as follows to minimize the quantization error:

$$\mathbf{Q}^{\star}(z) = \underset{\mathbf{Q}}{\operatorname{argmin}} \int p(z)(\mathbf{Q}(z) - z)^2 \, dz. \tag{4}$$

Given a number of decision intervals M, the optimal quantizer is expected to find the set of decision boundaries $\{b_q\}_0^M$ and quantized values $\{\hat{z}_q\}_1^M$. Solving the partial derivative of Eq.(4), we could have:

$$\hat{z}_q = \frac{\int_{b_{q-1}}^{b_q} z p(z) \, dz}{\int_{b_{q-1}}^{b_q} p(z) \, dz} \quad ; \qquad b_q = \frac{1}{2}(\hat{z}_q + \hat{z}_{q+1}). \tag{5}$$

The optimal solutions of Eq.(5) can be easily solved by the Lloyd's algorithm [17], outputting the optimal quantizer $\mathbf{Q}$ with decision boundaries $\{b_q\}_0^M$ and quantized values $\{\hat{z}_q\}_1^M$.

With the obtained quantizer fixed, we can use Eq.(2) to fine-tune the encoder-decoder network by minimizing the $\ell 1$-norm error between input and reconstructed images. The updated encoder-decoder network can then be used to update the non-uniform optimal quantizer by solving Eq.(5). Such an alternative optimization process continues till the loss function in Eq.(2) converges. Finally, we have the network parameters $\{\Omega, \Theta\}$ and the quantizer parameters $\{b_q, \hat{z}_q\}$.

3.3. Entropy Coding

Once the quantization outputs of image representations are obtained, we train an entropy coding network, whose architecture is shown in Figure 3, to perform lossless entropy coding. The entropy network predicts the probability of the current symbol from its context. After the probability of each symbol is predicted, arithmetic encoding is used to encode it to a bitstream.

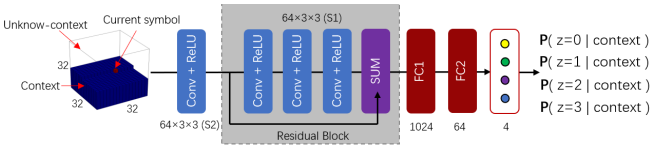

Fig. 3. Illustration of our entropy coding network.

The decision intervals M of the quantizer is set to 4 (2 bits). Therefore, the discrete representation coefficients $\hat{z}$ have only 4 probability values. To minimize the entropy of representation coefficients $\hat{z}$, we model this problem as a classification task (4 classes), and it can be trained with a cross-entropy loss function as follows:

$$\mathbf{H}(\hat{z}; \Pi) = -\frac{1}{C}\Big[\sum_{i=1}^{C}\sum_{j=0}^{3} 1\{\hat{z}^{(i)} = j\} \log(\mathbf{P}(\hat{z}^{(i)} = j | \hat{z}^{(i)}; \Pi))\Big], \tag{6}$$

where Π is the parameter set of the entropy network, C is the total number of element in $\hat{z}$, and $\mathbf{P}(\hat{z}^{(i)} = j | \hat{z}^{(i)}; \Pi)$ is the probability of the current symbol $\hat{z}^{(i)}$ (the i-th element of $\hat{z}$):

$$\mathbf{P}(\hat{z}^{(i)} = j | \hat{z}^{(i)}; \Pi) = \frac{e^{\Pi_j^T \hat{z}^{(i)}}}{\sum_{k=0}^{3} e^{\Pi_k^T \hat{z}^{(i)}}}. \tag{7}$$

To train the entropy network, we scan $\hat{z}$ in raster order to extract context blocks, each of which is composed of the available context, the current symbol $\hat{z}^{(i)}$ and the unavailable context. With the context blocks, we can minimize the entropy of representation coefficients $\hat{z}$ by maximizing the prediction accuracy of each symbol.

4. EXPERIMENTAL RESULTS

Dataset: We train the proposed deep compression network on the COCO test2017 dataset [18], the DIV2K dataset [19] and the Waterloo Exploration dataset [20]. Once the model is trained, we test it on the standard Kodak dataset [21], the McMaster dataset [22] and the BSD300 dataset [23].

Training Details: We crop the training data into 256×256 patches and make use of these patches to train the compression network. The mini-batch size is set to 15. The number of latent representation feature maps n is set to $\{32, 24, 16, 12, 6\}$ for different compression rates, and the number of quantized values is set to 4 (2 bits). The Adam solver [24] is adopted to optimize the network parameters $\{\Theta, \Omega, \Pi\}$. The learning rate starts from $1e{-}4$ and is then fixed to $1e{-}5$ when the training error stops decreasing. The training is terminated when the training error does not decrease in five sequential epochs. For the other hyper-parameters of Adam, we utilize the default setting.

The network is trained in CAFFE [25] with an Nvidia Titan Xp GPU. We experimental found that the alternative opti-

(a) **Original** (b) **JPEG** (BPP: 0.221 PSNR: 25.34db) (c) **JPEG2000** (BPP: 0.186 PSNR: 26.82db)

(d) **Ballé** (BPP: 0.175 PSNR: 27.04db) (e) **Our-Uniform** (BPP: 0.171 PSNR: 26.74db) (f) **Proposed** (BPP: 0.181 PSNR: 27.19db)

Fig. 4. Compression results by different methods at a compression rate around 0.2bpp (Image from Kodak dataset).

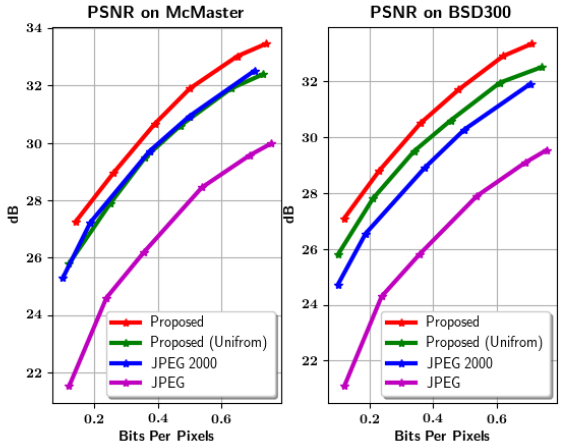

Fig. 5. Comparisons of the rate-distortion curves on McMaster dataset and BSD300 dataset.

mization process of our compressor will converge in less than 5 iterations.

Comparison methods: Most of the traditional compression methods (*i.e.*, JPEG and JPEG2000) are optimized in terms of PSNR. Therefore, we quantitatively evaluate different methods on PSNR for a fair comparison. Specifically, we compare our method with JPEG (using libjpeg [26]), JPEG2000 (using OpenJPEG [27]), the deep compressor Ballé [7] (the results are copied from [28]), and a baseline variant of our method (using the network of Figure 2 but with a uniform quantizer).

Note that since neither the source codes of recently published deep compressors [6, 7, 8, 10, 13] are available, nor the rate-distortion curves in terms of PSNR are reported, we can only compare with Ballé [7] which provides the compression results on the Kodak dataset on their website [28].

Comparison results: Figures 1 and 5 show the PSNR based rate-distortion curves on Kodak, McMaster and BSD300 datasets, respectively. Note that the curves for Ballé *et al.*'s method on McMaster and BSD300 are not available. One can see that the proposed iterative non-uniform quantization strategy can improve its baseline counterpart with uniform quantization. It also achieves better result than JPEG2000 and the recently developed deep compressor Ballé [7], and significantly outperforms the prevalent compressor JPEG.

Figure 4 compares the visual quality of compressed images *lighthouse* by the comparison methods at a compression rate around 0.2bpp. One can see that noticeable blocky and ringing artifacts are inevitable in the reconstructed images by traditional JPEG and JPEG2000 compression format. While Ballé and our baseline method can produce much better visual quality, they still blur much the edges and over-smooth the textures. Compared with these methods, the image compressed by our method is visually more pleasing.

5. CONCLUSION

Most of the existing deep network based image compressors employs a uniform quantizer which however limits their reconstruction accuracy. We presented an iterative non-uniform quantization scheme for deep image compression network. The *quantizer* and the *encoder-decoder* network were updated alternatively. With fixed quantizer, an encoder-decoder network was updated to minimize the ℓ_1 loss between the input and reconstructed images. While with fixed encoder-decoder network, the quantizer was optimized based on the distribution of encoded image coefficients. The alternative optimization process converged in five iterations in our experiments. Compared with JPEG2000 and previous deep compressors, our method exhibits better PSNR based rate-distortion curves on several benchmark test datasets.

6. REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPs*, 2012.

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[4] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *CVPR*, 2016.

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

[6] G. Toderici, D. Vincent, N. Johnston, S. Jin Hwang, D. Minnen, J. Shor, and M. Covell, "Full resolution image compression with recurrent neural networks," in *CVPR*, 2017.

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

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

[9] M. Li, W. Zuo, S. Gu, D. Zhao, and D. Zhang, "Learning convolutional networks for content-weighted image compression," *arXiv preprint arXiv:1703.10553*, 2017.

[10] O. Rippel and L. Bourdev, "Real-time adaptive image compression," *ICML*, 2017.

[11] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Van Gool, "Conditional probability models for deep image compression," *arXiv preprint arXiv:1801.04260*, 2018.

[12] N. Johnston, D. Vincent, D. Minnen, M. Covell, S. Singh, T. Chinen, S. J. Hwang, J. Shor, and G. Toderici, "Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks," *arXiv preprint arXiv:1703.10114*, 2017.

[13] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, and L. V. Gool, "Soft-to-hard vector quantization for end-to-end learning compressible representations," in *NIPs*, 2017.

[14] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–57, 2017.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *ICCV*, 2015.

[16] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient subpixel convolutional neural network," in *CVPR*, 2016.

[17] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014.

[19] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *CVPR Workshops*, 2017.

[20] K. Ma, Z. Duanmu, Q. Wu, Z. Wang, H. Yong, H. Li, and L. Zhang, "Waterloo exploration database: New challenges for image quality assessment models," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 1004–1016, 2017.

[21] Kodak, "http://r0k.us/graphics/kodak/," .

[22] L. Zhang, X. Wu, A. Buades, and X. Li, "Color demosaicking by local directional interpolation and nonlocal adaptive thresholding," *Journal of Electronic imaging*, vol. 20, no. 2, pp. 023016, 2011.

[23] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *ICCV*, 2001.

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

[25] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *ACM MM*. ACM, 2014, pp. 675–678.

[26] LibJPEG, "http://libjpeg.sourceforge.net/," .

[27] OpenJPEG, "http://www.openjpeg.org/," .

[28] NYU, "http://www.cns.nyu.edu/lcv/iclr2017/," .