Deep Image Compression via End-to-End Learning

Haojie Liu†, Tong Chen†, Qiu Shen, Tao Yue, and Zhan Ma
School of Electronic Science and Engineering, Nanjing University, Jiangsu, China
{haojie, tong}@smail.nju.edu.cn, {shenqiu, yuetao, mazhan}@nju.edu.cn
†*Authors contributed equally.*

Abstract

We present a lossy image compression method based on deep convolutional neural networks (CNNs), which outperforms the existing BPG, WebP, JPEG2000 and JPEG as measured via multi-scale structural similarity (MS-SSIM), at the same bit rate. Currently, most of the CNNs based approaches train the network using a l-2 loss between the reconstructions and the ground-truths in the pixel domain, which leads to over-smoothing results and visual quality degradation especially at a very low bit rate. Therefore, we improve the subjective quality with the combination of a perception loss and an adversarial loss additionally. To achieve better rate-distortion optimization (RDO), we also introduce an easy-to-hard transfer learning when adding quantization error and rate constraint. Finally, we evaluate our method on public Kodak and the Test Dataset P/M released by the Computer Vision Lab of ETH Zurich, resulting in averaged 7.81% and 19.1% BD-rate reduction over BPG, respectively.

1. Introduction

Images record the visual scene of our natural world and are often compressed for efficient network exchange and local storage. The most well known image compression algorithms are JPEG and its successors JPEG 2000. In the meantime, there are other alternatives such as WebP, BPG[1], and so on, shown quite impressive performance gains to further reduce the image size at the same quality. However, all of them present annoying artifacts (e.g., ringing, blocking, etc) at a high compression ratio, resulting in unpleasant user experience. With the exponential growth of the multimedia data, it is inevitable to develop another lossy image compression algorithms with higher performance (a.k.a., using tiny compressed size but presenting high-quality reconstruction).

Recent works have revealed the great potential in lossy image compression using deep learning [7, 2]. These methods utilize a single autoencoder or recurrent autoencoders to generate feature maps (fMaps) at the bottleneck layer for subsequent quantization and entropy coding. Quantization induced error/distortion would be utilized for end-to-end optimization.

For instance, Toderici *et al.* [7] have applied the Recurrent Neural Network (RNN) to produce entropy-coded bits progressively and to generate layered image reconstructions at different quality scales. Ballé *et al.* [2] have tried to optimize both distortion loss and entropy (rate) loss to improve the overall compression efficiency. In the meantime, Li *et al.* [5] have proposed a content-weighted approach to further optimize the compressed bit rates. All aforementioned methods have demonstrated the outstanding coding efficiency that outperform JPEG and JPEG2000 objectively and subjectively. Note that the objective metric utilized for distortion/quality evaluation is the multi-scale structural similarity (MS-SSIM) because of its superior correlation with the subjective opinion score.

On the other hand, Generative Adversarial Networks (GAN) and perceptual loss based approaches [4] have shown a great success in generating images with better visual quality. Instead of calculating distortions directly in pixel domain, these methods measure the similarity in high-level feature domain using a discriminator network or a pre-trained VGG network to mimic the discriminative characteristics of the human visual system (HVS).

Therefore, in this work, we have proposed a deep CNN, deep residual network specifically [3], based image compression scheme that optimizes the end-to-end rate-distortion performance of image compression jointly. Overall structure is consisted of a forward encoder, a quantizer, a backward decoder, a rate-distortion optimization (i.e., rate estimation and distortion measurement) and a visual enhancement subsystems.

To ensure the fast convergence of the deep neural network, we train the network progressively via transfer learning, i.e., using the networked trained at light compression (lower quantization) to learn the network at higher com-

[1]An image compression method uses the modified HEVC Intra Profile.

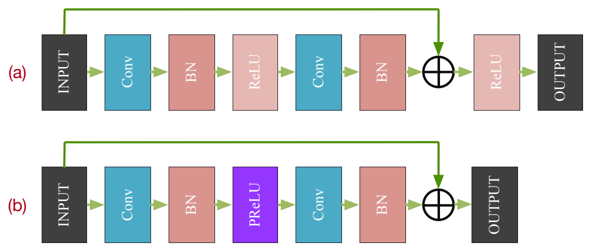


Figure 1. Illustration of a residual unit, (a) the default architecture [3]. (b) our proposed architecture with ReLU modifications.

pression ratio (higher quantization). In the meantime, motivated by the aforementioned perceptual enhancements using GAN and VGGnet, we have also introduced the perception loss and adversarial loss into the end-to-end optimization pipeline to generate texture and sharp details for noticeable visual quality improvement.

Compared with BPG (with input source sampled at YUV 4:2:0), our method has presented an impressive performance improvement with averaged 7.81% BD-Rate reduction (i.e., BD-Rate is measured using the MS-SSIM and Bits Per Pixel) on Kodak dataset, and averaged 19.1% on the Test Dataset P/M released by the Computer Vision Lab of ETH Zurich.

2. End-to-End Learning Framework

We utilize the deep residual network (ResNet) [3] in our framework because of its superior efficiency and fast convergence. To achieve a better convergence speed, we replace the default rectified linear units (ReLU) with the parametric rectified linear units (PReLU) and remove the non-linear mapping after the short connection for each residual unit, as shown in Fig. 1.

Overall, the forward encoder network contains eight residual units as shown in Fig. 2 and all the down-sampled operations are using a stride-2 4×4 convolutional layer. The decoder has a symmetrical architecture to reconstruct the signal from the compressed fMaps. We choose the pixel-shuffle layers as up-sampled operations considering its decent performances in super resolution (SR). The other convolutional layers all have 3×3 kernel size except the first and last layer using 5×5.

In addition, we use a rate estimation module to approximate the derivable rate loss for back propagation during the training step. The amount of the output features of encoder determine the upper limit of the rate. To adapt images with different content, we dynamically control the bit rates by using different network models. In one word, we apply less compression on images (i.e., more fMaps) with rich details and vice versa.

2.1. Quantization and Entropy Coding

A simple scalar quantization is employed first to reduce the number of bits for representing the extracted fMaps in encoder. First, we scale all the floating coefficients to 6-bit integer via,

$$X_Q = \texttt{Round}(X_E * (2^Q - 1)), \tag{1}$$

where $X_E \in (0, 1)$ represents the coefficient value of fMaps after the sigmoid activation, Q is the quantization level and set to 6. Then we applied PAQ (a lossless entropy coding method) for the quantized feature cofficients X_Q to generate the binary stream. Then the de-quantized feature coefficients $X_Q/(2^Q - 1)$ is fed into the decoder to finally reconstruct the image signals. We just skip the `Round` function during backpropagation.

2.2. Rate Estimation

The actual bitrates depend on the entropy of the quantized feature maps. We propose to apply the Lagrangian optimization framework to jointly consider the rate loss L_R and l-2 distortion loss. Here the rate loss L_R is defined as

$$L_R = -\mathbb{E}[\log_2 P_q], \tag{2}$$

where L_R is the entropy approximation of the fMaps at bottleneck layer. Since the derivatives of the quantization function are almost zero, we apply a piecewise linear approximation of the discrete P_q to ensure it continuous and differentiable. Note that Ballé [2] also applied similar idea to do joint rate and distortion optimization.

2.3. Network Training

Here, we present more details on how to train CNNs used in the work. In practice, we use the open source data sets released by the Computer Vision Lab of ETH Zurich in CLIC competition. All the images in the training sets are split into 128x128 patches randomly with a data augmentation method such as rotation or scaling, resulting in 80000 patches in total. The objective of training is to minimize the following loss function:

$$L = \frac{1}{N}\sum_{n=1}^{N} ||Y_n - X_n||^2 + \lambda L_R, \tag{3}$$

where X_n is the input image, Y_n is the decoded image, N represents the batch size. We introduce the parameter λ to control the penalty of rate loss L_R which is generated from the rate estimation module as shown in Eq. (2). Inspired by transfer learning, we also apply an easy-to-hard learning method mentioned in the deblocking method named AR-CNN and first set λ to 0. Without any rate control, we use the optimizer Adam (an adaptive learning rate method) with

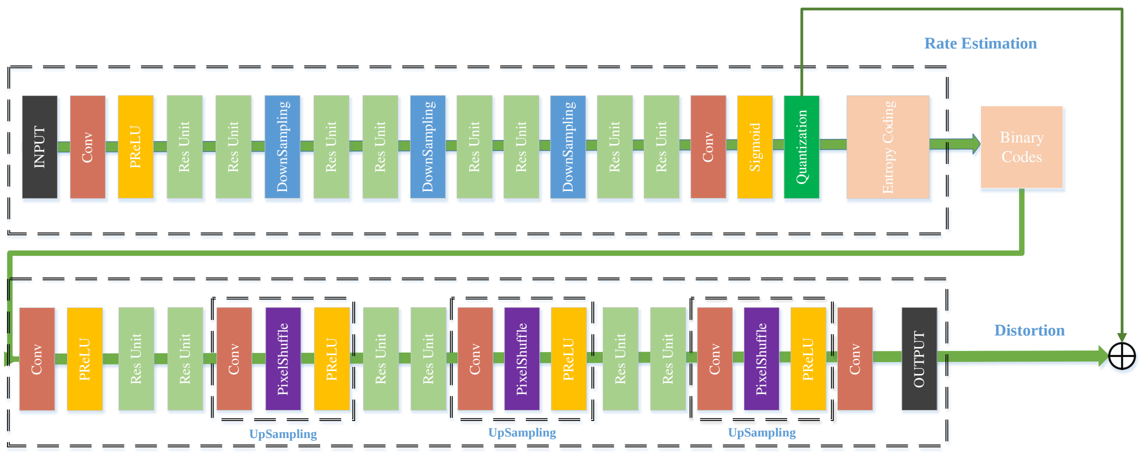


Figure 2. The entire end-to-end learning framework is consisted of a encoder, a quantizer, a decoder and a rate-distortion optimization subsystem with rate estimation and distortion measurement.

the learning rate 0.0001 to make fast convergence and generate the pre-training model first after 100 epochs. Then we increase the value of λ with a interval of 0.0001 from 0 to 0.002 every 5 epochs to progressively improve the level of rate constrain. Finally, it generates a lot of models with different rate and distortion to make a sophisticated RD optimization as shown in Fig. 3.

2.4. Visual Enhancement

We use another two loss functions to improve the subjective quality of the reconstructed image especially at low bit rate.

2.4.1 Perception Loss

Previous works have showed that optimizing the distortion in the feature domain can obviously increase the perceptual information. We choose the last convolutional layer of the 31-layer VGGnet [6] for feature extraction. The following Eq. 4 define the perception loss:

$$L_{percept} = \frac{1}{N}\sum_{n=1}^{N} ||\Psi(Y_n) - \Psi(X_n)||^2, \tag{4}$$

where Ψ is the VGGnet to compute the features.

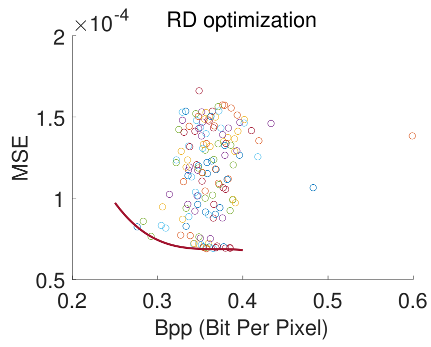


Figure 3. Illustrations of the different rate and distortion generated by different λ with a fixed amount of features. We can do a rate-distortion optimization (RDO) on these discrete points to get the optimized red curve that covers all points on its up-right side.

2.4.2 Adversarial Loss

We introduce another loss following the spirit of GANs. Then, a discriminated neural network D is established to distinguish whether a image is real or fake. For easier training, we replace the DCGAN with the improved Wasserstein GAN (WGAN) [1] to achieve faster convergence and more stable performance.

The WGAN uses an Earth-Move divergence to measure the similarity of two probabilities and enforce the generator to generate more realistic images. We add the new measurement into the loss function to encourage the reconstructed image having a high probability:

$$L_{generator} = -D(Y_n), \tag{5}$$

The D loss is defined in Eq. (6):

$$L_{discriminator} = D(Y_n) - D(X_n) + \beta L_{penalty}, \tag{6}$$

where D is the discriminative neural network, $L_{penalty}$ is the penalty term mentioned in the improved WGAN, β is the parameter of the $L_{penalty}$. Here we set it to 10.

In the end, we merge the different loss functions to build the final measurement component:

$$L_{final} = L2 + \lambda_1 L_R + \lambda_2 L_{percept} + \lambda_3 L_{generator}, \tag{7}$$

with λ_2 = 0.003 and λ_3 = 0.0001.

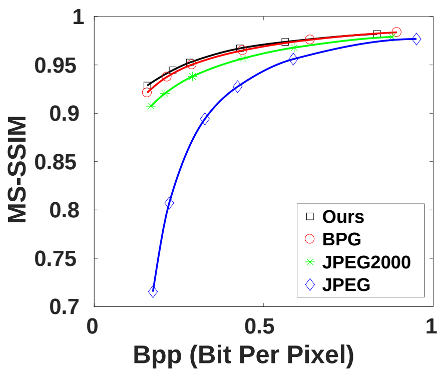
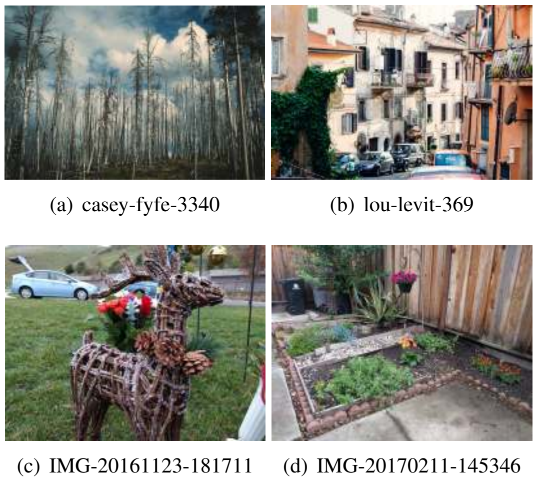


Figure 4. Compression performance on Kodak Dataset, measured in RGB domain, compared with JPEG, JPEG2000 and BPG


(a) casey-fyfe-3340 (b) lou-levit-369

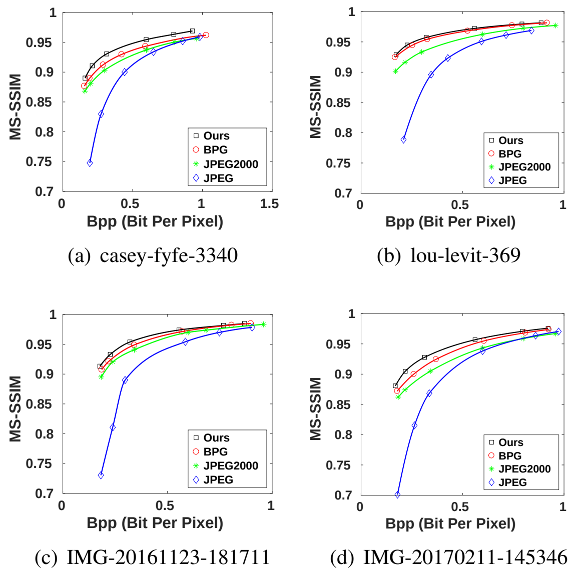
(c) IMG-20161123-181711 (d) IMG-20170211-145346

Figure 5. Four images sampled from CLIC test dataset

3. Performance Evaluation

We evaluate our performance on the dataset released by CLIC and Kodak PhotoCD data set, and compare with existing codecs including JPEG, JPEG2000, and BPG. Fig. 4 shows MS-SSIM performance over all 24 Kodak images and achieves averaged 7.81% BD-Rate reduction over BPG[2]. Moreover, we have more impressive performance on CLIC test dataset. We select four typical images with different types from the dataset as test samples, as shown in Fig. 5. Finally, the BD-Rate has separately reduced by 33.54%, 9.65%, 13.31% and 19.96%, as illustrated in Fig. 6 As can be seen from the results, our approach outperforms BPG and JPEG2000 for both overall performance and separate comparison using individual test image.

[2]Given that BPG demonstrates the state-of-the-art coding efficiency, we mainly present the comparison against it.



(a) casey-fyfe-3340 (b) lou-levit-369

(c) IMG-20161123-181711 (d) IMG-20170211-145346

Figure 6. Compression performance on four images, all measured in RGB domain, compared with JPEG, JPEG2000 and BPG.

4. Conclusion

An end-to-end learning framework based deep image compression scheme is detailed in this work, with innovations among residual unit, content adaptive fMaps, Lagrangian optimized rate-distortion adaptation, linear piecewise rate estimation, image visual quality enhancement with adversarial loss and perceptual loss included, and so on. Our network coder is trained using the public data set released by CLIC2018. Simulations are performed on independent images, resulting in impressive gains over the BPG and JPEG2000, both objectively and subjectively.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

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

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[4] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

[5] 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.

[6] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[7] G. Toderici, D. Vincent, N. Johnston, S. J. Hwang, D. Minnen, J. Shor, and M. Covell. Full resolution image compression with recurrent neural networks. In *CVPR*, pages 5435–5443. IEEE, 2017.