End-to-end Optimized Image Compression with Attention Mechanism

Lei Zhou [1],Zhenhong Sun [1],Xiangji Wu[1],Junmin Wu[1]
[1]Tucodec Inc
{zhoulei}@tucodec.com

Abstract

We present an end-to-end trainable image compression framework for low bit-rate and transparent image compression. Our method is based on variational autoencoder, which consists of a nonlinear encoder transformation, a soft quantizer, a nonlinear decoder transformation and an entropy estimation module. The prior probability of the latent representations is modeled by combining a hyperprior autoencoder and a Pixelcnn++ based context module and they are trained jointly with the transformation autoencoder with attention mechanism. In order to improve the compression performance, a non-local convolution based attention mechanism is designed for allocating bits adaptively. Finally, a novel rate allocation algorithm based on linear optimization is used to assign the bits for each image dynamically, considering the bits constraint of the challenge. Across the experimental results on validation and test sets, the optimized framework can generate the highest PSNR and MS-SSIM for low bit-rate compression competition, and cost the lowest bytes for transparent 40db competition.

1. Introduction

Recently, artificial neural networks (ANNs) have been applied to solve the image compression problem and a number of works have been proposed [3, 4, 10, 11, 8, 2, 6, 7, 5]. The previous methods can be divided into two categories. The first type of methods focus on generating superior perceptual quality [8, 6]. The basic idea for those methods is to generate high compression level without severe perceptual loss by generating image components, such as textures or plain regions. The generated components do not highly affect the the perceptual quality of the reconstructed images. Although the reconstructed images seem to be realistic, some details of the images may be modified. Another type of methods focus on designing end-to-end optimized image compression framework[11, 3, 4, 10, 7, 5]. In these approaches, the modules such as transformation, quantization and entropy estimation are optimized jointly. In [11], a certain proportion of binary latent representations is selected for information compression in every iteration, and the additional latent representations are increasingly exploited to achieve a progressive improvement in quality of the reconstructed images iteratively. Different from [11], current popular frameworks [3, 10, 4, 7, 5] formulate the image compression problem as being how to generate discrete latent representations with entropy as low as possible in an unsupervised way. In summary, the intuition of those methods such as [11] focus on how to generate as high-quality reconstructed images as possible given a fixed number of representations. On the other hand, the second kind of methods pay attention to reducing the entropy of latent representations. Hence rate or entropy estimation modules are crucial in an end-to-end optimized framework. In [3, 10], the entropy models with a fixed distribution for each representation are studied in the proposed novel frameworks, various entropy models such as gaussian mixture models, piecewise linear functions are designed for rate estimation. Their performance capabilities have been proven by comparing the results with those of conventional codecs such as JPEG2000. Motivated by the characteristics of natural images that the scales of the representations vary together in adjacent areas, [4] introduces a hyperprior based entropy model that estimates the scale of the gaussian distribution for each representation. The hyperprior based framework outperforms all previous ANN-based approaches, and approaches the most excellent traditional codec BPG. Two latest methods [7, 5] have outperformed BPG on both PSNR and MS-SSIM distortion metrics by predicting both mean and scale with context models. In [7], the authors extend the the works of [4] by generalizing the GSM model to a conditional Gaussian mixture model (GMM) and a masked convolution based context model is combined with the hyperprior to predict both the mean and scale of distribution. Similar to the idea in [7], [5] exploits two types of contexts, bit-consuming contexts and bit-free contexts, which allow the model to estimate the distribution of each latent representation more accurately with a more generalized form of the approximation models.

The proposed image compression framework is built on our CLIC 2018 solution[14]. Motivated by the above latest methods, our submitted solutions highlight three principal improvements: attention mechanism, soft quantization and context model with Pixelcnn++. The attention mechanism is designed to grasp information of larger scope, so as to improve the compression performance by allocating more bits to hard image components. The soft quantization is used to reduce round-off loss and improve the reconstruction quality. Pixelcnn++ [9] is applied to build the context model by capturing the long-range connections between latent features. Moreover, a resource allocation algorithm is designed to select the best compression parameter for each image considering the 0.15 bpp and 40db PSNR(0.993 MS-SSIM) constraints in the low bit-rate and transparent compression challenges.

2. End-to-End Optimized Image Compression

2.1. Encoder and Decoder with Attention Mechanism

The encoder-decoder architecture is similar to our CVPR 2018 CLIC framework [14]. An autoencoder with unbalanced structure as shown in Figure 1 is used. The encoder f_e and decoder f_d are composed of convolutions and GDN/IGDN nonlinearities. The GDN/IGDN implement a type of local divisive normalization transformation that has been proven to be particularly suitable for density modeling and images compression [3, 4]. In the encoder, a pyramidal feature fusion structure is proposed to learn optimal, nonlinear features for each scale. The features of intermediate layers with $\frac{1}{2}$, $\frac{1}{4}$ and $\frac{1}{8}$ of the original size are downsampled to $\frac{1}{16}$ via convolutions. In order to decrease the model parameters and reduce the computational costs, we have replaced all 5×5 convolutions used in [3, 4] with 3×3 convolutions. In order to capture the global dependencies between features, the residual non-local attention block (RNAB) [13] is integrated into the whole framework. RNAB is constructed by stacking several residual local and non-local attention blocks which are used to extract features that capture the long-range dependencies between pixels and pay more attention to the challenging parts. The detailed architecture of RNAB is displayed in Figure 1 as well. Then the downsampled features are concatenated, and a RNAB and a 1×1 convolution are applied to generate the encoded representation y. In decoder, convolutions, IGDN and RNAB are combined to reconstruct the image from quantized representation $\hat{y}$.

2.2. Soft Quantization

The round operation $\hat{y}_i = round(y_i)$ is widely used in learning based compression framework currently [4, 5, 7].

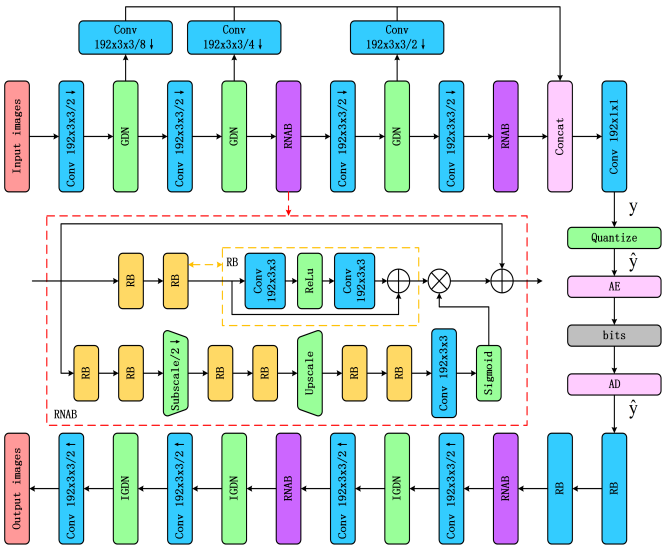

Figure 1. Illustration of the variational autoencoder architecture used in this paper. Convolution parameters are denoted as number of filter × kernel height × kernel width/ down or upsampling stride, where ↓ indicates downsampling and ↑ indicates upsampling. AE, AD represent arithmetic encoder and arithmetic decoder. RNAB stands for the residual non-local attention block.

However, the transformation from floating point numbers to integers can decrease the reconstruction quality significantly (at least 0.5db for PSNR and 1.5db for MS-SSIM). The soft quantization [2, 6] is integrated into our framework to decrease the round-off loss. Given the learnable clustering centers $C = \{c_1, ..., c_M\}$, the nearest neighbor assignments can be used to compute the latent representation in the forward pass:

$$\hat{y}_i = Q(y_i) = arg\ min_j \|y_i - c_j\|. \tag{1}$$

However, the above equation is differentiable, it's replaced by the soft assignment in the backward pass to compute the gradients:

$$\tilde{y}_i = \sum_{j=1}^{M} \frac{exp(-\sigma\|y_i - c_j\|)}{\sum_{l=1}^{M} exp(-\sigma\|y_i - c_l\|)} c_j. \tag{2}$$

As stated in [6], the soft quantization can be implemented in TensorFlow as

$$\hat{y}_i = tf.stopgradient(\hat{y}_i - \tilde{y}_i) + \tilde{y}_i. \tag{3}$$

2.3. Rate Estimation Module

We model each latent $\hat{y}_i$ as a Laplacian distribution with mean and scale parameters μ_i, σ_i convolved with a unit uni-

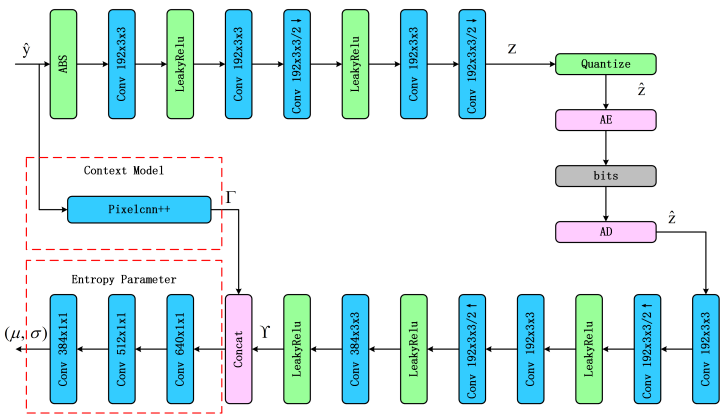

Figure 2. Illustration of the rate estimation module which consists of a hyperprior autoencoder, a context model and an entropy parameter sub-network.

form distribution. This ensures a good match between encoder and decoder distributions of both the quantized latents. Following the work of [7], both the hyperprior as well as the causal context of each latent $\hat{y}_i$ to predict the Laplacian parameters. As shown in Figure 2, the rate estimation module is consisted of three subnetworks: a hyperprior network H with parameter Θ_h, a context model T_{cm} with parameter Θ_{cm} and a entropy parameter subnetwork T_{ep} with parameter Θ_{ep}. The predicted Laplacian parameters are functions of learned parameters Θ_h, Θ_{cm} and Θ_{ep}:

$$p_{\hat{y}}(\hat{y}|\hat{z}, \Theta_h, \Theta_{cm}, \Theta_{ep}) = \prod_i (Lap(\mu_i, \sigma_i^2) * U(-\frac{1}{2}, \frac{1}{2}))(\hat{y}_i), \tag{4}$$

where $\mu_i, \sigma = T_{ep}(\Gamma, \Upsilon; \Theta_{ep})$ is the output of the entropy parameter subnetwork. $\Gamma = h_d(\hat{z}; \Theta_h)$ is the output of hyperprior network and $\Upsilon = T_{cm}(\hat{y}_{<i}; \Theta_{cm})$ is generated by the context model.

Hyperprior Network H: As illustrated in Figure 2, the subsampled feature y is fed into the hyperprior encoder which summarizes the distribution of standard deviations in $z = h_e(y)$. z is then quantized $\hat{z} = Q(z)$, compressed and transmitted as side information. The decoder estimates the parameter $\Gamma = h_d(\hat{z})$ and Γ is taken as the input of entropy parameter subnetwork T_{ep}. As to the distribution of $\hat{z}$, we model it as a non-parametric and fully factorized density model because there doesn't exist prior knowledge for $\hat{z}$, , similar to the strategy used in [4]:

$$p_{\hat{z}|\psi}(\hat{z}|\psi) = \prod_i (P_{z_i|\psi_i}(\psi_i) * \mu(-\frac{1}{2}, \frac{1}{2}))(\hat{z}_i), \tag{5}$$

where the vector ψ_i represents the parameters of each univariate distribution $P_{z_i|\psi_i}$.

Context Model:As to the T_{cm}, the Pixelcnn++ [9] is used to generate the context features Υ in our implementation. Different from the masked convolution used in [5, 7], the contexts in the proposed framework are conditioned on the left and to the up pixels in an image. Techniques such as downsampling and short-cut connection are used to increase the receptive field. Specially, the additional gated ResNet blocks with 1×1 convolution are inserted between regular convolution blocks to grow the receptive field. The experimental results show that Pixelcnn++ can improve the perceptual quality of generated images by encouraging the context model to capture long range dependencies.

Entropy Parameter sub-network T_{ep}: Then the outputs Γ and Υ are concatenated and fed into T_{ep}. The final layer of T_{ep} must have exactly twice as many channels as the bottleneck, so as to predict two values: the mean and scale of a Laplacian distribution for each latent. (Please refer to Figure 2 for the details of T_{ep})

Finally, the compression rates are composed of two part: rate R_y of compressed representation $\hat{y}$ and rate R_z of compressed side information $\hat{z}$. These rates are defined as follows:

$$\begin{aligned} R_y &= \sum_i -log_2(p_{\hat{y}}(\hat{y}|\hat{z}, \Theta_h, \Theta_{cm}, \Theta_{ep})), \\ R_z &= \sum_i -log_2(p_{\hat{z}_i|\psi}(\hat{z}|\psi)) \end{aligned} \tag{6}$$

2.4. Optimized Rate Control

Rate-Distortion optimization is a common strategy in algorithms such as HEVC and JPEG2000. Considering the bits constraint, a rate control optimization problem is defined to allocate the bits more effectively for each image:

$$min_{j \in M} \sum_{i=1}^{N} D_j(x_i, \hat{x}_i) \;\; st. \sum_i R_j^i < R_{max}, \tag{7}$$

where D represents the distortion between original image x_i and the reconstructed image $\hat{x}_i$. M is the vector set which contains all possible quality configurations for the set of images. N is the image number. D_j and R_j are the distortions and rates under configuration j. The best quality configuration is selected for each image via optimizing Eq (7) in our implementation. The rate control problem is optimized using dynamic programming algorithm.

3. Experimental Results

For training, 5000 high-quality images licensed under creative commons were downloaded from flickr.com and selected from CLIC 2019 challenge training set. These images were downsampled to 2000×2000 pixels and saved as lossless PNGs to avoid compression artifacts. From these

Table 1. Evaluation results on CLIC 2019 validation and test datasets.

	Methods	**PSNR**	**MS-SSIM**	**bytes**	**bpp**	**Decoding Time**
Validation	TucodecSSIM	29.840	0.9760	4692810	0.14906	23953868
	TucodecPSNR	32.520	0.9640	4722141	0.14999	15255275
	TucodecPSNR40dB	40.000	0.9930	27216543	0.86450	24013782
Test	TucodecSSIM	28.605	0.9739	15748980	0.15000	74252895
	TucodecPSNR	31.217	0.9575	15748347	0.14999	46174994
	TucodecPSNR40dB	40.000	0.9931	105429323	1.00415	75283641

downloaded images, we extracted two million patches with size 256×256 to train the network. Our team have submitted three solutions: TucodecPSNR, TucodecSSIM and TucodecPSNR40db. The results for the validation and test sets are reported in Table 1. The cluster number is set as 200 in the soft-quantization. We use two kinds of distortion measures in our solutions: mean square error and perceptional loss to train the autoencoder

$$L = \lambda D + R_y + R_z, \tag{8}$$

In TucodecSSIM which focuses on perception quality, the loss $D = 0.2 \times ||x - \hat{x}||_2^2 + 0.8 \times (1 - L_{msssim})$ is defined for the perceptional loss where L_{msssim} is as defined in [12]. Then the perceptional loss is combined with the same GAN setup defined in [8] for network optimization. Then five models with λ=0.2/0.3/0.4/0.5/0.6 are trained for rate control. Once the resource allocation is done, MS-SSIM of 0.976 and 0.974 can be achieved for validation and test sets respectively under the constraint of less than 0.15 bpp. In TucodecPSNR40db, the MSE loss $D = ||x - \hat{x}||_2^2$ is used for parameters learning and five models with λ=4096/4800/5500/6500/8000 are trained for rate control. Finally, the compressed files with bpp 0.864 and 1.00 are generated for validation and test sets given the at least 40 dB (aggregated) PSNR and at least 0.993 (aggregated) MS-SSIM constraints. TucodecPSNR is built on our modified version of H266 [1], the results for multiple QPs are generated for rate control. Furthermore, a post-processing module similar to the one used in [14] is designed to remove the compression artifacts and PSNR with 32.52 and 31.22 are obtained with 0.15 bpp.

4. Conclusion

In this paper, a novel deep learning based image compression framework with attention mechanism is designed for CLIC 2019 challenge. In the autoencoder part, an attention mechanism based on non-local convolution is integrated into the encoder-decoder procedure to capture the global connections between features in different channels and spatial locations. The experiments show that the attention mechanism can improve the compression performance by allocating more bits to important area in an unsupervised way. Moreover, our experiments have demonstrated that the soft quantization strategy can improve the reconstruction quality by decreasing the round-off loss, together with the Pixelcnn++ based contexts and hyperpriors. As shown in the results of the challenges on the validation set , our approaches TucodecPSNR and TucodecSSIM rank the 1st place in Low-rate compression for best PSNR and best MS-SSIM. The submitted method TucodecPSNR40db generate the lowest total bitrate in Transparent compression.

References

[1] H266 (https://de.wikipedia.org/wiki/h.266/), 2018. 4

[2] 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 *Advances in Neural Information Processing Systems*, pages 1141–1151, 2017. 1, 2

[3] J. Ballé, V. Laparra, and E. P. Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016. 1, 2

[4] 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. 1, 2, 3

[5] J. Lee, S. Cho, and S.-K. Beack. Context-adaptive entropy model for end-to-end optimized image compression. *arXiv preprint arXiv:1809.10452*, 2018. 1, 2, 3

[6] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Van Gool. Conditional probability models for deep image compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4394–4402, 2018. 1, 2

[7] D. Minnen, J. Ballé, and G. D. Toderici. Joint autoregressive and hierarchical priors for learned image compression. In *Advances in Neural Information Processing Systems*, pages 10771–10780, 2018. 1, 2, 3

[8] O. Rippel and L. Bourdev. Real-time adaptive image compression. *arXiv preprint arXiv:1705.05823*, 2017. 1, 4

[9] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017. 2, 3

[10] L. Theis, W. Shi, A. Cunningham, and F. Huszár. Lossy image compression with compressive autoencoders. *arXiv preprint arXiv:1703.00395*, 2017. 1

[11] 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 *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5435–5443. IEEE, 2017. 1

[12] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on*, volume 2, pages 1398–1402. Ieee, 2003. 4

[13] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu. Residual non-local attention networks for image restoration. *arXiv preprint arXiv:1903.10082*, 2019. 2

[14] L. Zhou, C. Cai, Y. Gao, S. Su, and J. Wu. Variational autoencoder for low bit-rate image compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2617–2620, 2018. 2, 4