Distortion scalable learned image compression

Renam C. da Silva
Multimedia R&D
Samsung Research Brazil (SRBR)
Campinas, Brazil
renam.cs@samsung.com

Vanessa Testoni
Multimedia R&D
Samsung Research Brazil (SRBR)
Campinas, Brazil
vanessa.t@samsung.com

***Abstract*—This work investigates a distortion scalable learned image encoder that, differently from other works, only encodes those most prominent feature maps that are found to benefit rate-distortion performance. This way, better suiting the texture complexity of content being coded and providing a competitive distortion scalable solution. Experimental results show that the proposed solution performs comparably to HEVC Intra with respect to the perceptually-inspired structural similarity (SSIM) index while being able to be optimized to suit a particular application.**

***Index Terms*—distortion scalability, learned image compression, autoencoder**

I. Introduction

The increasing availability of data, advances in electronic circuits dedicated to parallel processing of large data blocks and specialized computational libraries have been enabling a rapid evolution in machine learning. Researchers and practitioners are able to promptly experiment with new ideas and produce experimental results in order to drive research. Deep neural networks (DNN) [1] have been powering remarkable achievements on several challenging problems in natural language processing, computer vision, and speech recognition [2]. DNN are comprised of layered units which are able to learn a particular behavior by adjusting their parameters in order to minimize a loss function. Due to their ability to finding underlying structures in data [2], DNN soon were considered for dimensionality reduction with applicability in classification, visualization, communication and storage of high dimensional data [3], paving the way for image compression by means of end-to-end optimization frameworks [4]–[11].

In the last few years, an increasing number of research works has attempted to leverage advances in deep learning to bring new approaches and insights to image and video compression field. In this regard, we may observe mainly two work directions: *a)* approaches intended to fit into traditional hybrid predictive and transform coding framework (the Joint Video Experts Team has established ad hoc Group to investigate the benefit of deep learning technology in video compression [12]); and *b)* end-to-end optimization frameworks. This work is focused on the latter approach.

978-1-7281-1817-8/19/$31.00 ©2019 IEEE

II. End-to-end learned image compression

Motivated by the need for efficient compression methods targeting low spatial resolution images (thumbnails), Toderici *et al.* [4] investigated several neural network architectures [13]. Their models are comprised of an encoder E, a quantization function Q, and a decoder D that are optimized end-to-end. To enable progressive reconstruction at decoder side, the coding process proceeds iteratively in a given number of steps. A fundamental issue in learned image compression is how to model the quantization step, this is so because the derivative of the quantization function is either zero or infinite, what in turn make useless learning methods driven by gradient. To overcome this difficulty, most research works use differentiable approximations for the quantization function. Toderici *et al.* model the quantization as an additive noise process. In [5] Toderici *et al.* conducted further investigations on recurrent networks for image compression targeting full resolution images. To achieve competitive performance on images of arbitrary size, they attempted to design an improved residual patch encoder by experimenting with different recurrent units and deeper models. In addition to that, they introduced a second recurrent network for estimating the probability of a given feature map element conditioned by neighboring elements, this way enabling to capture long-term dependency within feature maps of the bottleneck layer.

Ballé *et al.* proposed a framework for end-to-end optimization of nonlinear analysis and synthesis transforms for image compression [7]. The transforms are learned by neural network models comprised of cascaded convolutional and generalized divisive normalization (GDN) layers. The analysis transform produces a latent representation for a given vector of pixel intensities. After quantizing the latent representation, an approximation of the input vector is obtained by applying the synthesis transform. In order to enable learning, the quantization function is replaced by a continuous approximation during training. Ballé *et al.* have advanced their baseline framework [8]–[10], [14]. Their most recent work [14] presents tools for exploiting the spatial structure within the latent representation to achieve more accurate probability models. For this purpose, additional pieces of information are encoded aiming at reducing the uncertainty of the latent representation.

Theis *et al.* [6] investigated the use of convolutional layers structured in residual blocks. To get around the quantization

issue, they replaced the derivative of the quantization function with the derivative of a smooth approximation. Leveraging the architecture [8], Dumas *et al.* [11] investigated different strategies for achieving different rate-distortion operational points. They showed that one may train a single model by fixing the support region for the additive noise (replacing quantization) and setting the Lagrangian multiplier to a value enough to give the model capacity to reach high rates. At inference time, different rate-distortion operational points would be achieved merely varying the quantization step size. This contrasts to previous works that would require several models to be trained in order to reach different rate-distortion operational points.

Rippel and Bourdev [15] used autoencoder-based pyramidal models that are able to explicitly copy with image structures both within a single scale and shared across scales. To this end, the input image is subjected to several rounds of transforming and down-sampling operations. The resulting sets of coefficients are subjected to inter-scale alignment for joint processing. In turn, the resulting tensor is subjected to quantization and entropy coding. Another novelty introduced in [15] is the use of adversarial training [16] to optimize the models using a reconstruction loss combined with a discriminator loss. In the context of generative adversarial networks [16], the whole autoencoder (encoder and decoder) plays the role of the generator network whose objective is to produce images that the discriminator is unable to distinguish from the target images. The generator and discriminator networks are put to compete with each other. The payoff for this adversarial training is visually pleasing reconstructed images.

III. Distortion Scalable Learned Image Compression

The ability of scalable decoding is of paramount importance in several imaging applications. A single compressed bitstream providing resolution scalability would allow devices equipped with different displaying capability to decode nothing more than a portion of the bitstream required to fill their screens. Additionally, a scalable bitstream may provide distortion (or SNR) scalability. In this case, starting from a base layer reconstruction, increasingly more accurate reconstructions would be obtained by decoding chunks of the encoded bitstream. This is well appreciated, for instance, in content delivery over best-effort networks, in which case a server would store a single encoded bitstream and would be able to deliver it in a scalable fashion. In the context of traditional image coding, the JPEG2000 standard [17], [18] generates highly scalable bitstream.

In this work, we focus on distortion scalability by leveraging previous works on learned image compression. Different from other learning-based solutions, our distortion scalable learned image encoder only encodes those most prominent feature maps that benefit rate-distortion performance, allowing scaling a trained model to the texture and structural complexity of the content being encoded. To drive feature maps selection for different rate-distortion trade-offs, we derive an empirical relationship between the quantization step size and the Lagrangian

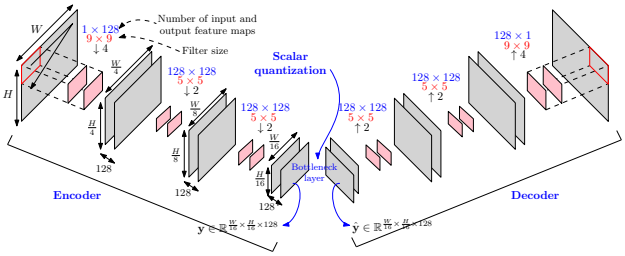


Fig. 1. The strided convolutional layers in the encoder are followed by GDN layers. Similarly, the transposed convolutional layers in the decoder are followed by inverse GDN layers. ↓ indicates down-sampling as a result of strided convolutions, whereas ↑ indicates up-sampling resulting from transposed convolutions. $a \times b$ indicates the number of input and output feature maps, whereas $c \times c$ indicates the filter spatial extent.

multiplier used to weight rate and distortion during coding process. Moreover, to achieve more compression efficiency, the selected feature maps are entropy encoded using arithmetic coding combined with context-adaptive models to exploit spatial structure in them. The compressed bitstream enables distortion scalable decoding. Each selected and conveyed feature map may be (arithmetic) decoded and fed to the decoder part of the autoencoder to produce a reconstruction for the input image.

A. Model architecture

Fig. 1 depicts schematically the autoencoder architecture for learning-based image compression. In fact, we leverage the autoencoder architecture introduced in [8], [11]. The autoencoder possess a straightforward architecture comprised of strided convolutional layers and generalized divisive normalization (GDN) layers in the encoder, whereas the decoder is comprised of transposed convolution and inverse GDN layers (for short IGDN).

The objective is to train jointly in an end-to-end fashion, an encoder $g_e(\cdot, \boldsymbol{\theta})$, parametrized by $\boldsymbol{\theta}$, and a decoder $g_d(\cdot, \boldsymbol{\phi})$, parametrized by $\boldsymbol{\phi}$, to minimize a Lagrangian cost function J, which is a trade-off of rate and distortion. The set of learned parameters $\boldsymbol{\theta}$ includes all convolutional kernels, biases and GDN parameters. Likewise, the set of learned parameters $\boldsymbol{\phi}$ includes all transposed convolutional kernels, biases and IGDN parameters. During training, to circumvent the issue of quantizing feature maps $\mathbf{y}$ and to allow learning, we follow the same strategy as [8], [11] and replace scalar quantization (step size 1) by a continuous approximation, effectively performed by additive random noise, that is, $\tilde{\mathbf{y}} = \mathbf{y} + \Delta\mathbf{y}$, with $\Delta y_i \sim u(-1/2, 1/2)$. For the purpose of rate estimation, the probability density function (pdf) $p(\tilde{\mathbf{y}})$ is simply regarded as fully factorized $p(\tilde{\mathbf{y}}) = \prod_{i=1}^{\frac{W}{16}\cdot\frac{H}{16}\cdot m} p(y_i)$ with elements in each feature map sharing a common probability model. During training, the pdf of each feature map is fitted by a piecewise linear function [8], [11]. Putting all those pieces together, the objective function becomes:

$$\min_{\boldsymbol{\phi},\boldsymbol{\theta}} \mathbb{E}\Big[d(\mathbf{x}, \hat{\mathbf{x}}) + \lambda \cdot \sum_{i}^{\frac{H}{16}\cdot\frac{W}{16}\cdot m} \log_2 p_{y_i}(y_i + \Delta y_i)\Big] \qquad (1)$$

where $\hat{\mathbf{x}} = g_d(g_e(\mathbf{x};\boldsymbol{\theta}) + \Delta\mathbf{y};\boldsymbol{\phi})$, d is the distortion and $\mathbb{E}$ is expectation (in practice approximated by averaging over a large image dataset).

1) Training: To achieve different operational rate-distortion points, a common practice in learning-based image compression solutions is to train several models targeting different rate-distortion trade-offs [8], [9]. Dumas *et al.* [11] have shown that one could train a single model and still be able to achieve different rate-distortion trade-off points by simply varying the quantization step size at inference. Following this finding, we train a single model with Lagrangian multiplier $\lambda = 10000.0$ and an additive uniform random noise with unitary support; $\Delta y_i \sim u(-1/2, 1/2)$. The distortion measure between the input image $\mathbf{x}$ and its reconstruction $\hat{\mathbf{x}} = g_d(g_e(\mathbf{x};\boldsymbol{\theta})+\Delta\mathbf{y};\boldsymbol{\phi})$ is simply the mean-squared error (MSE). Training is conducted over a subset of the Imagenet dataset [19]. Precisely, over an image dataset comprised of random luminance crops (spatial resolution 256×256) taken from 7000 images. The training images are arranged in batches of 10 and training is carried out for 2858 epochs using gradient descent with initial learning rate 10^{-4}. It takes approximately 30 hours to train using a single GPU.

B. Distortion scalability

An inconvenience of the aforementioned model is that the number of feature maps at the bottleneck layer is fixed in a way to give the model enough capacity to learn from a large number of images (exhibiting different structures and textures complexities). During inference, the filters responses in the feature maps $\mathbf{y}$ will reflect the structures present in the input image. Due to this fact, the feature maps will contribute differently, in a content-dependent way, to the reconstruction of the input image. To allow scaling a trained model to the content being encoded, a subset of the feature maps may be selected according to a rate-distortion criterion. To this end, the decoder is embedded in the encoder and each feature map at the bottleneck layer is evaluated whether or not benefits rate-distortion performance, an index for the selected feature map is encoded in the bitstream along with the filter responses, thus giving rise to a distortion scalable bitstream. Notice that as soon a feature map is available, the decoder part of the autoencoder is able to obtain a reconstruction for the input image.

The feature maps at the bottleneck are first sorted in decreasing order of energy of their elements, that is, $\sum_{j=1}^{\frac{W}{16}\cdot\frac{H}{16}} y_{j,m_1}^2 \leq \cdots \leq \sum_{j=1}^{\frac{W}{16}\cdot\frac{H}{16}} y_{j,m_{128}}^2$. The idea is to prioritize feature maps with strong filter responses. After that, the encoder evaluates the sorted feature maps, one at a time, by checking for reduction in a Lagrangian cost function of the rate and distortion. That is, for each feature map, the elements are scalar quantized with quantization step size Δ as detailed in Section III-B1. The rate required to entropy encode the quantized feature map is estimated using the context-adaptive models described in Section III-B2. The quantized feature map is then fed to the decoder to produce a reconstruction for the input image.

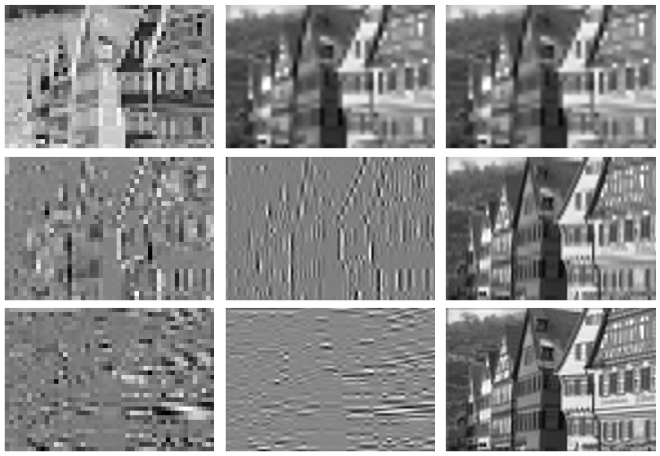


Fig. 2. *left)* decoded feature maps; *middle)* reconstructions obtained from feature map on the left; *right)* reconstructions obtained by accumulating feature maps.

The distortion between the obtained reconstruction and the original image is added to the rate term weighted by the Lagrangian multiplier λ_s. If the feature map is found to reduce the Lagrangian cost (initialized with a large value), it is encoded in the bitstream and kept for next iterations, then the minimum Lagrangian cost is updated. Otherwise, it is discarded and the next feature map is evaluated. The idea is to select and encode only those most prominent feature maps that are found to benefit rate-distortion performance.

Fig. 2 illustrates the reconstructions obtained after selecting three feature maps. The decoded feature maps are shown in the left column. The corresponding reconstructions are shown in the middle, this is the reconstruction obtained solely considering the feature maps in the left. Whereas in the right column are shown the reconstructions obtained by accumulating the feature maps. It is interesting to notice the information conveyed by each feature map. For instance, one may say that the first feature map conveys a 'sort of dc information'. As matter of fact, we noticed that, if this feature map were omitted, the obtained reconstruction would be nothing more than a gradient-looking image, no matter how much quality we put in other feature maps. Furthermore, the second and the third selected feature maps have added noticeable details to object boundaries (both vertical and horizontal transitions) in the reconstructed images.

1) Lagrangian multiplier and quantization step size: At inference, when a trained model is actually used for compressing a given image, the continuous relaxation used during training is replaced by a scalar uniform quantization. The elements within each feature maps are first clipped to a minimum and maximum value and then centered by subtracting a mean value. The result is then scalar quantized as:

$$\hat{y}_i = \mathrm{round}\left(\frac{y_i}{\Delta}\right) \cdot \Delta \tag{2}$$

where $\mathrm{round}\,(y_i/\Delta)$ rounds its arguments to the nearest integer and gives the quantization index q_{y_i} For simplicity,

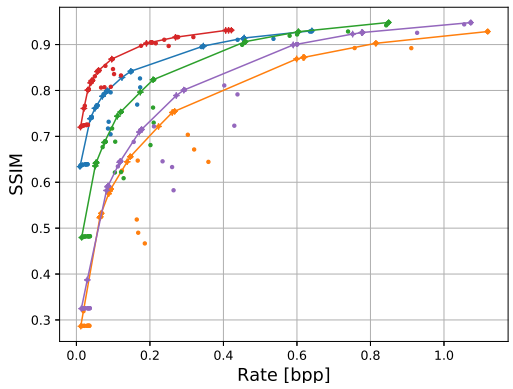


Fig. 3. Rate-SSIM points resulted by setting different values to λ_s and Δ. The values of λ_s and Δ giving rise to those points at the convex hull are gathered and used to fitting a function. Each color is a different image.

the same Δ is used for all feature maps.

To determine the value of the Lagrangian multiplier λ_s to weight the distortion and rate terms in the course of selecting those feature maps to be encoded, we run the distortion scalable encoder for several values of λ_s and Δ and gather those values leading to the points lying in the convex hull, as shown in Figure 3. After repeating this procedure for a set of images[1], we found a linear function by fitting the gathered set of points $(\Delta, \log_{10}(\lambda_s))$. The resulted function relating the Lagrangian multiplier and the quantization step size is given by:

$$\lambda_s = 10^{a \cdot \Delta + b}, \text{ with } a = 0.23005 \text{ and } b = -3.99265 \quad (3)$$

2) Arithmetic coding: During training, we assumed a simplistic fully factorized pdf for feature maps at the bottleneck layer. However, as one may notice in Fig. 2 (left column), the feature maps exhibit non-negligible structure. Nearby values within each feature maps tend to be coupled together, notably in the first feature map. To achieve more compression efficiency, we exploit such structure by using context-adaptive models combined with arithmetic coding. The quantization indexes for each feature map are entropy encoded with respect to a set of conditional probability mass functions (pmf), which are estimated by means of a prediction by partial matching (ppm) [20] scheme of order o. That is, by looking o previously encoded quantization indexes (the so-called context) in order to estimate the probability for the current quantization index. If in the course of estimating the probability for a quantization index, a given context has never occurred in the model of order o, *i.e.* $p_m(q_{y_i}|q_{y_{i-1}}, ..., q_{y_{i-o}})$, the model order is lowered until the given context is found. In the worst case, an unconditional model with even probability is used.

An advantage of this scheme is that entropy coding gets better as the models are updated and becomes suited to the image being encoded. In order to boost entropy coding efficiency, the models of each feature map are initialized with models learned after compressing a large set of images[1]. The estimated conditional probabilities for the quantization indexes are used by arithmetic coder [21] to actually generate a compressed bitstream to enable a standalone decoder to decode it. After experimenting with the order of ppm, we found that $o = 2$ would be a good compromise between compression efficiency and memory consumption.

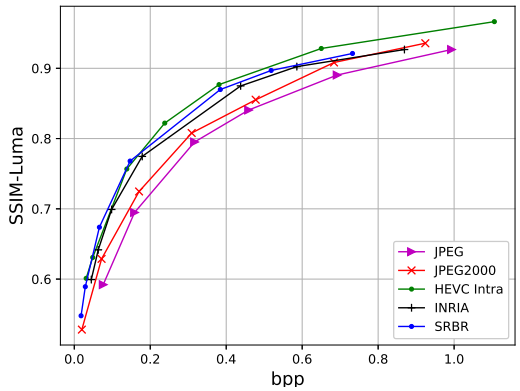
IV. Experimental results

Performance assessment is carried out comparatively to JPEG[2], JPEG2000[3], HEVC Intra[4], and INRIA [11] using the Kodak dataset[5], which is comprised of 24 images of spatial resolutions 768×512 or 512×768. The experiments were conducted only for *luma* component. The reconstruction quality is evaluated with respect to PSNR and SSIM [22]. In order to produce results for JPEG, the quality parameter was set to $\{5; 10; 20; 30; 50; 70\}$. Results for JPEG2000 were obtained by setting the target quality to $\{24; 26; 28; 30; 32; 34; 36\}$. In the case of HEVC Intra, the quantization parameter was set to $\{25; 30; 34; 37; 40; 45; 47\}$.

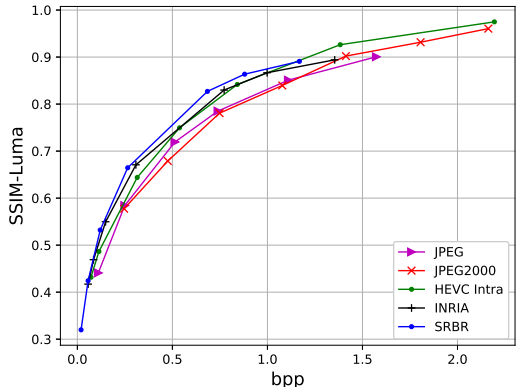
After training the proposed learning-based scalable codec (named SRBR) using the training steps described in Section III-A, the experimental results are obtained by varying the quantization step size used for quantizing the feature maps at the bottleneck layer; namely $\Delta = \{1.0; 2.0; 4.0; 6.0; 8.0; 10.0\}$. All selected feature maps were quantized with the same quantization step size (Equation 2). For a given quantization step size, the Lagrangian multiplier λ_s used to compute the Lagrangian cost in the course of selecting those feature maps worth to be encoded is computed by Equation 3.

Fig. 4 shows the rate-distortion performance for two images from Kodak dataset, namely *kodim06* and *kodim13*. As one may observe, the distortion scalable learned codec consistently outperforms JPEG and JPEG2000 with respect to perceptually-inspired measure SSIM. This is also the case regarding the non-scalable INRIA which encodes feature maps irrespective of their contributions to rate-distortion performance and relies on estimated entropy for rate computation. On the other hand, SRBR encodes feature maps driven a Lagrangian cost of the rate and distortion, this way adapting to the content being encoded. This is done by generating a 'tangible' compressed bitstream providing distortion scalability, from which a standalone decoder may produce increasingly less distorted reconstructions. Regarding the heavily optimized HEVC Intra, the distortion scalable learned codec presents comparable performance, notably at the lower bit rates. Although quite common in the learned image compression literature, it is debatable if is adequate drawing conclusions from averaged results. Despite that, in order to summarize the performance assessment, Fig. 5 shows the rate-distortion results obtained

[1]BSD500 training set: eecs.berkeley.edu/Research/Projects/CS/vision/
[2]libjpeg available at https://jpeg.org/jpeg/software.html
[3]OpenJPEG available at https://www.openjpeg.org/
[4]HM v. 16.9 available at: https://hevc.hhi.fraunhofer.de/
[5]http://r0k.us/graphics/kodak/

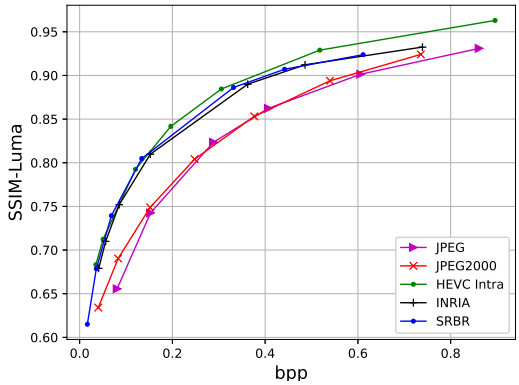


(a) *Kodim 06*

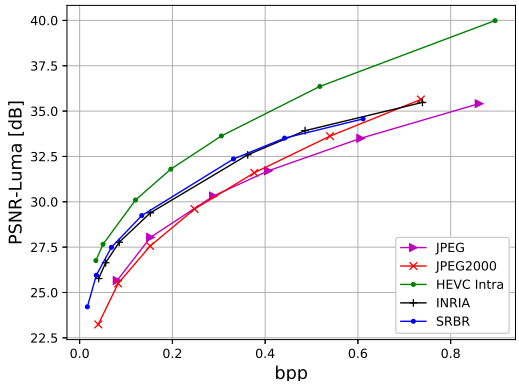


(b) *Kodim 13*

Fig. 4. Rate-SSIM curve for two examples



(a) Reconstruction quality assessed using SSIM



(a) Reconstruction quality assessed using PSNR

Fig. 5. Average performance over Kodak dataset

averaging over all images of the Kodak dataset, both for SSIM and PSNR as quality measures. Compared to HEVC, SRBR shows to be much more competitive when the quality score is assessed using SSIM. Perhaps the localized connectivity pattern of the convolutional filters may favor structures preservation. Naturally, this is not a demerit for SRBR as SSIM is known to be better predict the perceived quality. Also, HEVC is heavily optimized for MSE, besides being the product of decades of technical amelioration. Regarding execution time in CPU, the SRBR encoder takes approximately 50 seconds to compress a single image from Kodak dataset, whereas the decoder takes 13 seconds.

Table I presents the average bit rate savings (in percentage) of the proposed SRBR with respect to JPEG, JPEG2000, and INRIA. SRBR provides considerable bit rate saving regarding INRIA, showing that the strategy of selecting and coding only the most prominent feature maps (in a content-dependent way) enable better bit rate allocation. On average, SRBR provides 40.25%, 12.18% and 8.7% bitrate savings with respect to JPEG, JPEG2000 and INRIA, respectively.

V. Conclusions

This work investigates a distortion scalable learned image compression solution. At inference, driven by a Lagrangian cost of rate and distortion, a single trained encoder selects its most prominent generated feature maps for coding. This way, scaling the trained model to the content being encoded and providing distortion scalability as each encoded feature map improves upon the reconstruction based on previously selected features maps. To better suit the content being encoded, improving compression performance, a context-adaptive entropy coding scheme is devised to exploit structure within each feature map.

Acknowledgment

Results presented in this work were obtained during the Deep Codec project funded by Samsung Eletrônica da Amazônia Ltda under the Brazilian Informatics Law 8.248/91.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Neworks," in *International Conference on Neural Information Processing Systems (NIPS)*, Lake Tahoe, USA, December 2012.

[2] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep Learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[3] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[4] George Toderici, Sean M. OMalley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell, and Rahul Sukthankar, "Variable Rate Image Compression with Recurrent Neural Networks," in *International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico, May 2016.

TABLE I
PSNR-BJONTEGAARD DELTA FOR THE PROPOSED SRBR WITH RESPECT TO JPEG, JPEG2000, AND INRIA

Image	JPEG BD-Rate	JPEG2000 BD-Rate	INRIA BD-Rate
kodim01	-38.15%	-21.61%	-12.06%
kodim02	-53.03%	-12.79%	-19.16%
kodim03	-49.32%	-14.46%	-11.05%
kodim04	-48.89%	-11.92%	-6.77%
kodim05	-37.90%	-19.86%	-2.91%
kodim06	-36.33%	-10.01%	-14.62%
kodim07	-42.95%	-23.32%	-5.66%
kodim08	-30.08%	-4.34%	-5.68%
kodim09	-40.33%	-9.34%	-7.21%
kodim10	-41.62%	-11.11%	-5.38%
kodim11	-39.90%	-14.18%	-7.77%
kodim12	-48.82%	-17.37%	-19.05%
kodim13	-23.76%	3.73%	-11.08%
kodim14	-39.04%	-17.33%	-4.04%
kodim15	-47.07%	-7.89%	-8.83%
kodim16	-50.14%	-21.19%	-18.24%
kodim17	-42.76%	-13.76%	-4.12%
kodim18	-28.35%	-6.56%	-1.12%
kodim19	-36.21%	-7.25%	-5.37%
kodim20	-42.16%	-8.95%	-11.72%
kodim21	-33.97%	-11.60%	-9.89%
kodim22	-39.78%	-18.80%	-5.80%
kodim23	-47.42%	-6.42%	-8.43%
kodim24	-28.13%	-5.92%	-2.94%
Aver.	**-40.25**	**-12.18**	**-8.70**

[5] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, and Joel Shor, "Full Resolution Image Compression with Recurrent Neural Networks," in *Computer Vision and Pattern Recognition (CVPR)*, Honolulu, USA, July 2017.

[6] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár, "Lossy Image Compression with Compressive Autoencoders," in *International Conference on Learning Representations (ICLR)*, Toulon, France, April 2017.

[7] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli, "End-to-end optimization of nonlinear transform codes for perceptual quality," in *Picture Coding Symposium (PCS)*, Nuremberg, Germany, December 2016.

[8] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli, "End-to-end Optimized Image Compression," in *International Conference on Learning Representations (ICLR)*, Toulon, France, April 2017.

[9] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston, "Variational Image Compression with a Scale Hyperprior," in *International Conference on Learning Representations (ICLR)*, Vancouver, Canada, April 2018.

[10] Johannes Ballé, "Efficient Nonlinear Transforms for Lossy Image Compression," in *Picture Coding Symposium (PCS)*, San Francisco, USA, June 2018.

[11] Thierry Dumas, Aline Roumy, and Christine Guillemot, "Autoencoder based Image Compression: Can the Learning be Quantization Independent?," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Canada, April 2018.

[12] S. Liu, B. Choi, K. Kawamura, Y. Li, L. Wang, P. Wu, and H. Yang, "JVET AHG9 report: Neural Networks in Video Coding (AHG9) ," July 2018, Doc. JVET-K0009-v1.

[13] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals, "Recurrent Neural Network Regularization," in *arXiv preprint arXiv:1409.2329v5*, February 2015.

[14] David Minnen, Johannes Ballé, and George Toderici, "Joint Autoregressive and Hierarchical Priors for Learned Image Compression," in *32nd Conference on Neural Information Processing System (NIPS)*, Montreal, Canada, December 2018.

[15] Oren Rippel and Lubomir Bourdev, "Real-Time Adaptive Image Compression," in *International Conference on Machine Learning (ICML)*, Sydney, Australia, August 2017.

[16] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative Adversarial Nets," in *International Conference on Neural Information Processing Systems*, Montreal, Canada, December 2014.

[17] Athanassios Skodras, Charilaos Christopoulos, and Touradj Ebrahimi, "The JPEG 2000 Still Image Compression Standard," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 36–58, September 2001.

[18] David S. Taubman and Michael W. Marcellin, "JPEG2000: Standard for Interactive Imaging," *Proceedings of the IEEE*, vol. 90, no. 8, pp. 1336–1357, August 2002.

[19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "ImageNet: A Large-scale Hierarchical Image Database," in *IEEE Conference on Computer Vision and Pattern Recognition*, Miami, USA, June 2009.

[20] J. Cleary and I. Witten, "Data Compression Using Adaptive Coding and Partial String Matching," *IEEE Transactions on Communications*, vol. 32, no. 4, pp. 396–402, April 1984.

[21] Timothy C. Bell, John G. Cleary, and Ian H. Witten, *Text Compression*, Prentice Hall, 1 edition, 1990.

[22] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: from Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004.