Learned BI-Resolution Image Coding Using Generalized Octave Convolutions

Mohammad Akbari[1], **Jie Liang**[1], **Jingning Han**[2], **Chengjie Tu**[3]
[1] Simon Fraser University, Canada
[2] Google Inc., Mountain View
[3] Tencent Technologies
akbari@sfu.ca, jiel@sfu.ca, jingning@google.com, chengjietu@tencent.com

Abstract

Learned image compression has recently shown the potential to outperform the standard codecs. State-of-the-art rate-distortion (R-D) performance has been achieved by context-adaptive entropy coding approaches in which hyperprior and autoregressive models are jointly utilized to effectively capture the spatial dependencies in the latent representations. However, the latents are feature maps of the same spatial resolution in previous works, which contain some redundancies that affect the R-D performance. In this paper, we propose a learned bi-resolution image coding approach that is based on the recently developed octave convolutions to factorize the latents into high and low resolution components. Therefore, the spatial redundancy is reduced, which improves the R-D performance. Novel generalized octave convolution and octave transposed-convolution architectures with internal activation layers are also proposed to preserve more spatial structure of the information. Experimental results show that the proposed scheme outperforms all existing learned methods as well as standard codecs such as the next-generation video coding standard VVC (4:2:0) in both PSNR and MS-SSIM. We also show that the proposed generalized octave convolution can improve the performance of other auto-encoder-based schemes such as semantic segmentation and image denoising.

Introduction

Deep learning-based image compression has shown the potential to outperform standard codecs such as JPEG2000 and H.265/HEVC-based BPG (Bellard 2017). These approaches automatically discover and exploit the features of the data; thereby achieve better compression performance compared to traditional methods. Various learning-based image compression frameworks have been proposed in the last few years (Toderici et al. 2015; Ballé, Laparra, and Simoncelli 2016; Rippel and Bourdev 2017; Theis et al. 2017; Agustsson et al. 2017; Johnston et al. 2017; Minnen, Ballé, and Toderici 2018; Lee, Cho, and Beack 2018; Akbari, Liang, and Han 2019; Akbari et al. 2020; Li et al. 2019, 2020).

In (Ballé et al. 2018), a conditional Gaussian scale mixture (GSM)-based entropy model was introduced where the scale parameters were conditioned on a hyperprior learned using a hyper auto-encoder. The compressed hyper latents were transmitted and added to the bit stream as side information. This model was extended in (Minnen, Ballé, and Toderici 2018; Lee, Cho, and Beack 2018) where a Gaussian mixture model (GMM) with both mean and scale parameters conditioned on the hyperprior was utilized. In these methods, the hyperpriors were combined with autoregressive priors generated using context models, which outperformed BPG in terms of both PSNR and MS-SSIM. Another context-adaptive approach was introduced by (Zhou et al. 2019) in which multi-scale masked convolutional networks were utilized for their autoregressive model combined with hyperpriors.

The state of the art in learned image compression has been achieved by context-adaptive entropy methods in which hyperprior and autoregressive models are combined (Minnen, Ballé, and Toderici 2018). These approaches are jointly optimized to effectively capture the spatial dependencies and probabilistic structures of the latent representations, which lead to a compression model with superior rate-distortion (R-D) performance. However, similar to natural images, the latents are usually represented by feature maps of the same spatial resolution, which result in some spatial redundancies. These maps can be factorized into high and low resolution components with effective inter-resolution communication (Chen et al. 2019), which result in less spatial redundancies and consequently better R-D performance.

In this paper, a learned bi-resolution image coding and entropy model is introduced in which octave convolutions (Chen et al. 2019) are employed to factorize the latent representations into high resolution (HR) and low resolution (LR) components. As LR is represented by a lower spatial resolution, the corresponding spatial redundancy is reduced and the compression performance is improved, similar to wavelet transforms (Antonini et al. 1992). In addition, due to the effective communication between HR and LR components in octave convolutions, the reconstruction performance is also improved. In the original octave convolution (Chen et al. 2019), fixed interpolation methods are used for down- and up-sampling operations, which do not retain the spatial information. So, it can negatively affect the image compression performance. In order to preserve the spatial structure of the latents in our image coding framework, we develop novel generalized octave convolution and octave transposed-convolution architectures with internal activation layers.

Experiment results show that the proposed scheme outperforms all existing learning-based methods and standard

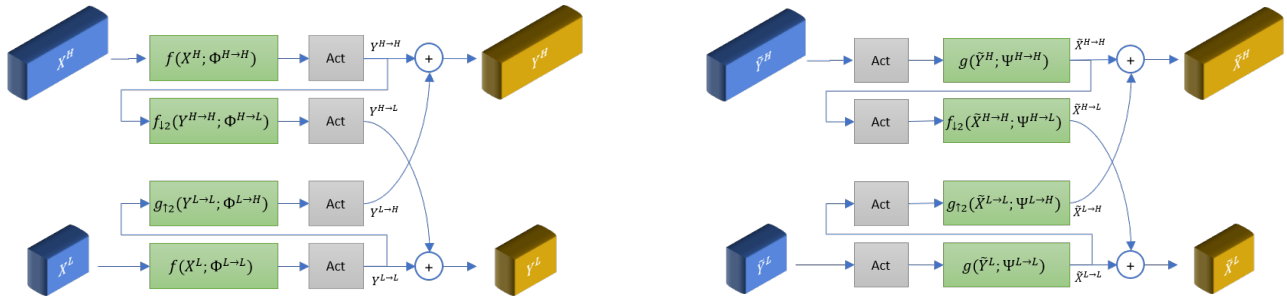


Figure 1. Architecture of the proposed generalized octave convolution (GoConv) shown in the left figure, and transposed-convolution (GoTConv) shown in the right figure. Act: the activation layer; f: regular vanilla convolution; g: regular transposed-convolution; $f_{\downarrow 2}$: regular convolution with stride 2; $g_{\uparrow 2}$: regular transposed-convolution with stride 2.

codecs in terms of both PSNR and MS-SSIM on the Kodak dataset. The framework proposed in this work bridges the wavelet transform and deep learning; therefore, many techniques in the wavelet transform can be used in the proposed framework to further improve its performance. This will have profound impact on future image coding research. In the supplementary material, we additionally show that the proposed generalized octave convolution and transposed-convolution architectures can improve the performance and the computational complexity in other auto-encoder-based computer vision tasks such as semantic segmentation and image denoising.

Generalized Octave Convolution

In the vanilla convolution, all input and output feature maps are of the same spatial resolution. As a result, there will be some unnecessary redundancies, which will hurt the performance in applications such as compression. To address this problem, in the recently developed octave convolution (Chen et al. 2019), the feature maps are factorized into HR and LR components with different resolutions, where each component is processed with different convolutions. As a result, the resolution of LR feature maps can be spatially reduced, which saves both memory and computation.

The factorization of input vector X in octave convolutions is denoted by $X = \{X^H, X^L\}$, where $X^H \in \mathbb{R}^{h\times w\times(1-\alpha)c}$ and $X^L \in \mathbb{R}^{\frac{h}{2}\times\frac{w}{2}\times\alpha c}$ are respectively the HR and LR maps. The ratio of channels allocated to the LR feature representations (i.e., at half of spatial resolution) is defined by $\alpha \in [0, 1]$. The factorized output vector is denoted by $Y = \{Y^H, Y^L\}$, where $Y^H \in \mathbb{R}^{h'\times w'\times(1-\alpha)c'}$ and $Y^L \in \mathbb{R}^{\frac{h'}{2}\times\frac{w'}{2}\times\alpha c'}$ are the output HR and LR maps.

In (Chen et al. 2019), intra-resolution update strategy is used to update the information within each HR and LR, while inter-resolution communication is performed to further enable information exchange between the two parts. As in filter bank theory (Vaidyanathan 2006), the octave convolution allows information exchange between the HR and LR feature maps. For the intra-resolution update, the vanilla convolution is used. However, up- and down-sampling interpolations are applied to compute the inter-resolution communication.

As reported in (Chen et al. 2019), due to the effective inter-resolution communications, the octave convolution can have better performance in classification and recognition performance compared to the vanilla convolution. Since the octave convolution allows multi-resolution feature maps, it is very suitable for image compression, which motivates us to apply it to learned image compression. However, some modifications are necessary in order to get a good performance.

In the original octave convolution, the average pooling and nearest interpolation are respectively employed for down- and up-sampling operations in inter-resolution communication (Chen et al. 2019). Such conventional interpolations do not preserve spatial information and structure of the input feature map. In addition, in convolutional auto-encoders where sub-sampling needs to be reversed at the decoder side, fixed operations such as pooling result in a poor performance (Springenberg et al. 2014).

In this work, we propose a novel generalized octave convolution (GoConv) in which strided convolutions are used to sub-sample the feature vectors and compute the inter-resolution communication in a more effective way. Fixed sub-sampling operations such as pooling are designed to forget about spatial structure, for example, in object recognition where we only care about the presence or absence of the object, not its position. However, if the spatial information is important, strided convolution can be a useful alternative. With learned filters, strided convolutions can learn to handle discontinuities from striding and preserve more spatial properties required in down-sampling operation (Springenberg et al. 2014). Moreover, since it can learn how to summarize, better generalization with respect to the input is achieved. As a result, better performance with less spatial information loss can be achieved, especially in auto-encoders where it is easier to reverse strided convolutions. Moreover, as in ResNet, applying strided convolution (i.e., convolution and down-sampling at the same time) reduces the computational cost compared to a convolution followed by a fixed down-sampling operation (e.g., average pooling).

The proposed GoConv scheme is shown in Figure 1. Compared to the original octave convolution, we apply another important modification regarding the inputs to the inter-resolution convolution operations. In order to calculate the inter-resolution communication outputs (denoted by $Y^{H\to L}$ and $Y^{L\to H}$), the input HR and LR vectors (denoted by X^H

and X^L) are considered as the inputs to the HR-to-LR and LR-to-HR convolutions, respectively ($f_{\downarrow 2}$ and $g_{\uparrow 2}$ in Figure 1). This strategy is only efficient for the stride of 1 (i.e., the size of input and output HR and LR vectors is the same). However, in GoConv, this can result in significant information loss for larger strides. As an example, consider using stride 2, which results in down-sampled output HR and LR feature maps (half resolution of input HR and LR maps). To achieve this, stride 2 is required for the intra-resolution convolution (f in Figure 1). However, for the inter-resolution convolution $f_{\downarrow 2}$, a harsh stride of 4 should be used, which results in significant spatial information loss.

To deal with this problem, we instead use two consecutive convolutions with stride 2 where the first convolution is indeed the intra-resolution operation f. In other words, to compute $Y^{H\to L}$, we exploited the filters learned by f to have less information loss. Thus, instead of X^H and X^L, we set $Y^{H\to H}$ and $Y^{L\to L}$ as inputs to $f_{\downarrow 2}$ and $g_{\uparrow 2}$. The output HR and LR feature maps in GoConv are then formulated as follows:

$$\begin{aligned} Y^H &= Y^{H\to H} + g_{\uparrow 2}(Y^{L\to L};\Phi^{L\to H}),\\ Y^L &= Y^{L\to L} + f_{\downarrow 2}(Y^{H\to H};\Phi^{H\to L}),\\ \text{with } Y^{H\to H} &= f(X^H;\Phi^{H\to H}),\\ Y^{L\to L} &= f(X^L;\Phi^{L\to L}), \end{aligned} \tag{1}$$

where $f_{\downarrow 2}$ and $g_{\uparrow 2}$ are respectively Vanilla convolution and transposed-convolution operations with stride of 2. $\Phi^{H\to H}$ and $\Phi^{L\to L}$ are intra-resolution and $\Phi^{L\to H}$ and $\Phi^{H\to L}$ are inter-resolution kernels.

In original octave convolutions, activation layers (e.g., ReLU) are applied to the output HR and LR maps. However, as shown in Figure 1, we utilize activations for each internal convolution performed in our proposed GoConv. In this case, we assure that the activation functions are properly applied to each feature map computed by convolution operations. Each of the inter- and intra-resolution components is then followed by an activation layer in GoConv.

We also propose a generalized octave transposed-convolution (GoTConv), which can replace the conventional transposed-convolution commonly employed in deep auto-encoder (encoder-decoder) architectures (Figure 1). Let $\tilde{Y} = \{\tilde{Y}^H, \tilde{Y}^L\}$ and $\tilde{X} = \{\tilde{X}^H, \tilde{X}^L\}$ respectively be the factorized input and output vectors, the output HR and LR maps $\tilde{X}^H$ and $\tilde{X}^L$) in GoTConv are obtained as follows:

$$\begin{aligned} \tilde{X}^H &= \tilde{X}^{H\to H} + g_{\uparrow 2}(\tilde{X}^{L\to L};\Psi^{L\to H}),\\ \tilde{X}^L &= \tilde{X}^{L\to L} + f_{\downarrow 2}(\tilde{X}^{H\to H};\Psi^{H\to L})\\ \text{with } \tilde{X}^{H\to H} &= g(\tilde{Y}^H;\Psi^{H\to H}),\\ \tilde{X}^{L\to L} &= g(\tilde{Y}^L;\Psi^{L\to L}), \end{aligned} \tag{2}$$

where $\tilde{Y}^H, \tilde{X}^H \in \mathbb{R}^{h\times w\times(1-\alpha)c}$ and $\tilde{Y}^L, \tilde{X}^L \in \mathbb{R}^{\frac{h}{2}\times\frac{w}{2}\times\alpha c}$. Unlike GoConv in which regular convolution operation is used, transposed-convolution denoted by g is applied for intra-resolution update in GoTConv. For up- and down-sampling operations in inter-resolution communication, the same strided convolutions $g_{\uparrow 2}$ and $f_{\downarrow 2}$ as in GoConv are respectively utilized. $\Psi^{H\to H}$ and $\Psi^{L\to L}$ are intra-resolution and $\Psi^{L\to H}$ and $\Psi^{H\to L}$ are inter-resolution kernels.

Similar to the original octave convolution, the proposed GoConv/GoTConv are designed and formulated as generic, plug-and-play units. As a result, they can respectively replace vanilla convolution and transposed-convolution units in any CNN architecture, especially auto-encoder-based frameworks such as image compression, image denoising, and semantic segmentation. When used in an auto-encoder, the input image to the encoder is not represented as a bi-resolution tensor. In this case, to compute the first GoConv layer output in the encoder, Equation 1 is modified as follows:

$$Y^H = f(X;\Phi^{H\to H}), \quad Y^L = f_{\downarrow 2}(Y^H;\Phi^{H\to L}), \tag{3}$$

Similarly, at the decoder side, the output of the last GoTConv is a single tensor representation, which can be formulated by modifying Equation 2 as:

$$\begin{aligned} \tilde{X} &= \tilde{X}^{H\to H} + g_{\uparrow 2}(\tilde{X}^{L\to L};\Psi^{L\to H}),\\ \text{with } \tilde{X}^{H\to H} &= g(\tilde{Y}^H;\Psi^{H\to H}),\\ \tilde{X}^{L\to L} &= g(\tilde{Y}^L;\Psi^{L\to L}). \end{aligned} \tag{4}$$

Compared to GoConv, the process of using activations for each internal transposed-convolution in GoTConv is inverted, where the activation layer is followed by inter- and intra-resolution communications as shown in Figure 1.

Octave-Based Bi-Resolution Image Coding

Octave convolution is similar to the wavelet transform (Antonini et al. 1992), since it has lower spatial resolution in LR than in HR, which can be used to improve the R-D performance in learning-based image compression frameworks. Moreover, due to the effective inter-resolution communication as well as the receptive field enlargement in octave convolutions, they also improve the performance of the analysis (encoding) and synthesis (decoding) transforms.

The overall architecture of the proposed bi-resolution image compression framework is shown in Figure 2. Similar to (Minnen, Ballé, and Toderici 2018), our architecture is composed of two sub-networks: the core auto-encoder and the entropy sub-network. The core auto-encoder is used to learn a quantized latent vector of the input image, while the entropy sub-network is responsible for learning a probabilistic model over the quantized latent representations, which is utilized for entropy coding.

In order to handle bi-resolution entropy coding, we have made several improvements to the scheme in (Minnen, Ballé, and Toderici 2018). First, all vanilla convolutions in the core encoder, and hyper encoder are replaced by the proposed GoConv, and all vanilla transposed-convolutions in the core and hyper decoders are replaced by GoTConv. In (Minnen, Ballé, and Toderici 2018), each convolution/transposed-convolution is accompanied by an activation layer. In our scheme, we move these layers into the GoConv/GoTConv architectures and directly apply them to the inter- and intra-resolution components. Generalized divisive normalization (GDN) and inverse GDN (IGDN) transforms are respectively used for the GoConv/GoTConv employed in the proposed deep encoder

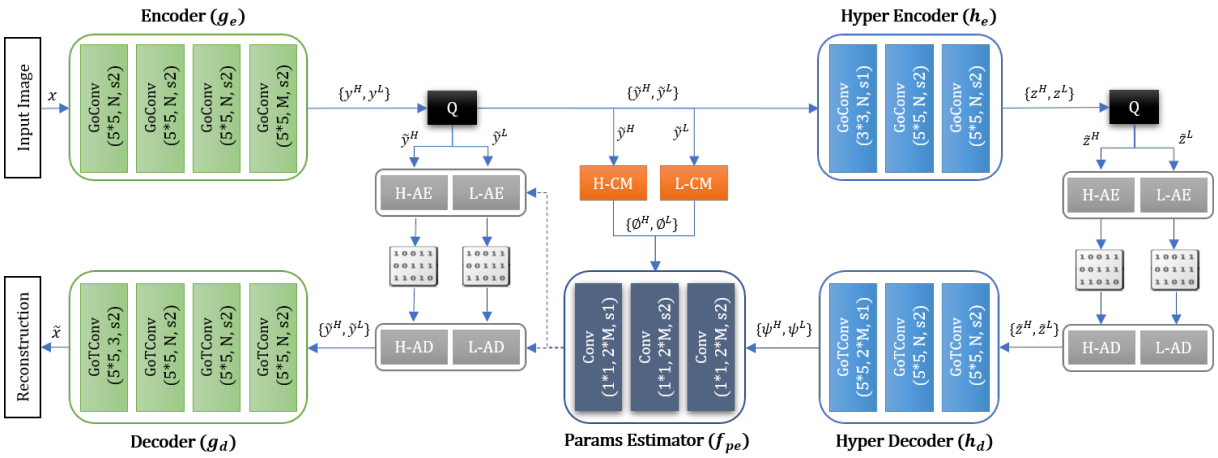


Figure 2. Overall framework of the proposed image compression model. H-AE and H-AD: arithmetic encoder and decoder for HR latents. L-AE and L-AD: arithmetic encoder and decoder for LR latents. H-CM and L-CM: the HR and LR context models each composed of one 5*5 masked convolution layer with 2*M filters and stride of 1. Q: represents the additive uniform noise for training, or uniform quantizer for the test.

and decoder, while Leaky ReLU is utilized for the hyper auto-encoder and the parameters estimator. The convolution properties (i.e., size and number of filters and strides) of all networks including the core and hyper auto-encoders, context models, and parameter estimator are the same as in (Minnen, Ballé, and Toderici 2018).

Let $x \in \mathbb{R}^{h \times w \times 3}$ be the input image, the bi-resolution latent representations are denoted by $\{y^H, y^L\}$ where $y^H \in \mathbb{R}^{\frac{h}{16} \times \frac{w}{16} \times (1-\alpha)M}$ and $y^L \in \mathbb{R}^{\frac{h}{32} \times \frac{w}{32} \times \alpha M}$ are generated using the parametric deep encoder (i.e., analysis transform) g_e represented as:

$$\{y^H, y^L\} = g_e(x; \theta_{ge}), \tag{5}$$

where θ_{ge} is the parameter vector to be optimized. M denotes the total number of output channels in g_e, which is divided into $(1-\alpha)M$ channels for HR and αM channels for LR (i.e., at half spatial resolution of the HR part). The calculation in Equation (3) is used for the first GoConv layer, while the other encoder layers are formulated using Equation (1).

At the decoder side, the parametric decoder (i.e., synthesis transform) g_d with the parameter vector θ_{gd} reconstructs the image $\tilde{x} \in \mathbb{R}^{h \times w \times 3}$ as follows:

$$\tilde{x} = g_d\left(\{\tilde{y}^H, \tilde{y}^L\}; \theta_{gd}\right) \text{ with } \{\tilde{y}^H, \tilde{y}^L\} = Q\left(\{y^H, y^L\}\right), \tag{6}$$

where Q represents the addition of uniform noise to the latent representations during training, or uniform quantization (i.e., round function in this work) and arithmetic coding/decoding of the latents during the test. As illustrated in Figure 2, the quantized HR and LR latents $\tilde{y}^H$ and $\tilde{y}^L$ are entropy-coded using two separate arithmetic encoder and decoder.

The entropy sub-network in our architecture contains two models: a context model and a hyper auto-encoder (Minnen, Ballé, and Toderici 2018). The context model is an autoregressive model over bi-resolution latent representations. Unlike the other networks in our architecture where GoConv are

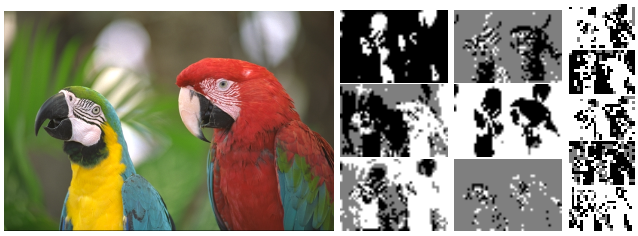


Figure 3. Sample HR and LR latent representations. Left column: original image; Middle columns: HR; Right column: LR.

incorporated for their convolutions, we use Vanilla convolutions in the context model to ensure that the causality of the contexts is not spoiled due to the intra-resolution communication in GoConv. The contexts of the HR and LR latents, denoted by ϕ_i^H and ϕ_i^L, are then predicted with two separate models f_{cm}^H and f_{cm}^L defined as follows:

$$\phi_i^H = f_{cm}^H(\tilde{y}_{<i}^H; \theta_{cm}^H) \text{ and } \phi_i^L = f_{cm}^L(\tilde{y}_{<i}^L; \theta_{cm}^L), \tag{7}$$

where θ_{cm}^H and θ_{cm}^L are the parameters to be generalized. Both f_{cm}^H and f_{cm}^L are composed of one 5*5 masked convolution (Van den Oord et al. 2016) with stride of 1.

The hyper auto-encoder learns to represent side information useful for correcting the context-based predictions. The spatial dependencies of $\{\tilde{y}^H, \tilde{y}^L\}$ are then captured into the bi-resolution hyper latent representations $\{z^H, z^L\}$ using the parametric hyper encoder h_e (with the parameter vector θ_{he}) defined as:

$$\{z^H, z^L\} = h_e\left(\{\tilde{y}^H, \tilde{y}^L\}; \theta_{he}\right). \tag{8}$$

The quantized hyper latents are also part of the generated bitstream that is required to be entropy-coded and transmitted. Similar to the core latents, two separate arithmetic coders

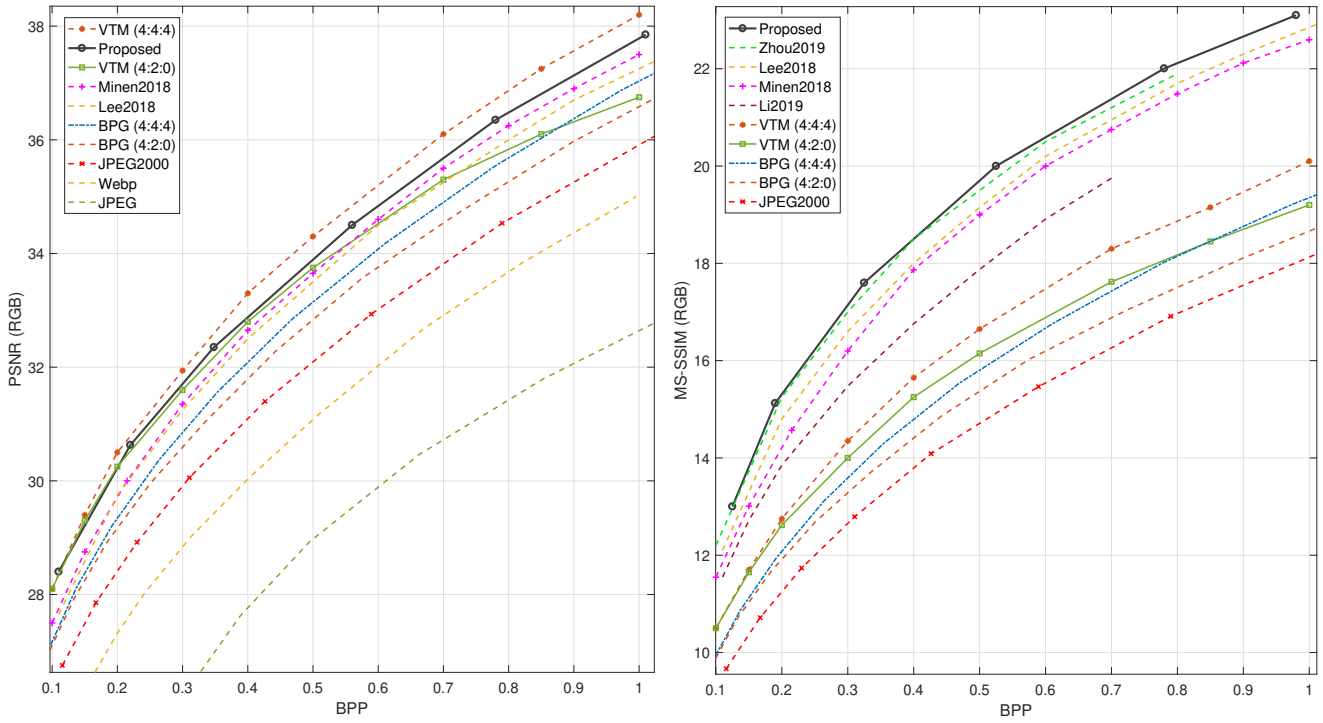


Figure 4. Kodak comparison results of our approach with traditional codecs and learning-based image compression methods.

are used for the quantized HR and LR $\tilde{z}^H$ and $\tilde{z}^L$. Given the quantized hyper latents, the side information used for the entropy model estimation is reconstructed using the hyper decoder h_d (with the parameter vector θ_{hd}) formulated as:

$$\begin{aligned} \{\psi^H, \psi^L\} &= h_d\left(\{\tilde{z}^H, \tilde{z}^L\}; \theta_{hd}\right) \\ \text{with } \{\tilde{z}^H, \tilde{z}^L\} &= Q\left(\{z^H, z^L\}\right). \end{aligned} \tag{9}$$

As shown in Figure 2, to estimate the mean and scale parameters required for a conditional Gaussian entropy model, the information from both context model and hyper decoder is combined by another networks, denoted by f_{pe}^H and f_{pe}^L (with the parameter vectors θ_{ep}^H and θ_{ep}^L), represented as follows:

$$\{\mu_i^H, \sigma_i^H\} = f_{pe}^H\left(\{\psi^H, \phi_i^H\}; \theta_{ep}^H\right), \tag{10}$$

$$\{\mu_i^L, \sigma_i^L\} = f_{pe}^L\left(\{\psi^L, \phi_i^L\}; \theta_{ep}^L\right), \tag{11}$$

where μ_i^H and σ_i^H are the parameters for entropy modelling of the HR information, and μ_i^L and σ_i^L are for the LR information.

The objective function for training is composed of two terms: rate R, which is the expected length of the bitstream, and distortion D, which is the expected error between the input and reconstructed images. The R-D balance is determined by a Lagrange multiplier denoted by λ. The R-D optimization problem is then defined as follows:

$$\begin{aligned} \mathcal{L} &= R + \lambda D \\ \text{with } R &= R^H + R^L, \\ D &= \mathbb{E}_{x\sim p_x}\left[d(x, \hat{x})\right], \end{aligned} \tag{12}$$

where p_x is the unknown distribution of natural images and D can be any distortion metric such as mean squared error (MSE) or MS-SSIM. R^H and R^L are the rates corresponding to the HR and LR information (bitstreams) defined as follows:

$$\begin{aligned} R^H &= \mathbb{E}_{x\sim p_x}\left[-\log_2 p_{\tilde{y}^H|\tilde{z}^H}(\tilde{y}^H|\tilde{z}^H)\right] \\ &\quad + \mathbb{E}_{x\sim p_x}\left[-\log_2 p_{\tilde{z}^H}(\tilde{z}^H)\right], \\ R^L &= \mathbb{E}_{x\sim p_x}\left[-\log_2 p_{\tilde{y}^L|\tilde{z}^L}(\tilde{y}^L|\tilde{z}^L)\right] \\ &\quad + \mathbb{E}_{x\sim p_x}\left[-\log_2 p_{\tilde{z}^L}(\tilde{z}^L)\right], \end{aligned} \tag{13}$$

where $p_{\tilde{y}^H|\tilde{z}^H}$ and $p_{\tilde{y}^L|\tilde{z}^L}$ are respectively the conditional Gaussian entropy models for HR and LR latent representations (y^H and y^L) formulated as:

$$\begin{aligned} &p_{\tilde{y}^H|\tilde{z}^H}(\tilde{y}^H|\tilde{z}^H, \theta_{hd}, \theta_{cm}^H, \theta_{ep}) = \\ &\prod_i \left(\mathcal{N}(\mu_i^H, \sigma_i^{2H}) * \mathcal{U}(-\tfrac{1}{2}, \tfrac{1}{2})\right)(\tilde{y}_i^H), \\ &p_{\tilde{y}^L|\tilde{z}^L}(\tilde{y}^L|\tilde{z}^L, \theta_{hd}, \theta_{cm}^L, \theta_{ep}) = \\ &\prod_i \left(\mathcal{N}(\mu_i^L, \sigma_i^{2L}) * \mathcal{U}(-\tfrac{1}{2}, \tfrac{1}{2})\right)(\tilde{y}_i^L), \end{aligned} \tag{14}$$

where each latent is modelled as a Gaussian convolved with a unit uniform distribution, which ensures a good match between encoder and decoder distributions of both quantized and continuous-values latents. The mean and scale parameters μ_i^H, σ_i^H, μ_i^L, and σ_i^L are generated via the networks f_{pe}^H and f_{pe}^L defined in Equations 10 and 11.

Since the compressed hyper latents $\tilde{z}^H$ and $\tilde{z}^L$ are also part of the generated bitstream, their transmission costs are

also considered in the rate term formulated in Equation 13. As in (Ballé et al. 2018; Minnen, Ballé, and Toderici 2018), to model HR and LR hyper-priors, we assume the entries to be independent and identically distributed (i.i.d.) and fit a univariate piecewise linear density model to represent each channel j. The non-parametric, fully-factorized density models for the HR and LR hyper latents are then formulated as follows:

$$p_{\tilde{z}^H|\Theta^H}(\tilde{z}^H|\Theta^H) = \prod_j \left(p_{z_i^H|\Theta_j^H}(\Theta_j^H) * \mathcal{U}(-\tfrac{1}{2}, \tfrac{1}{2})\right)(\tilde{z}_j^H),$$
$$p_{\tilde{z}^L|\Theta^L}(\tilde{z}^L|\Theta^L) = \prod_j \left(p_{z_i^L|\Theta_j^L}(\Theta_j^L) * \mathcal{U}(-\tfrac{1}{2}, \tfrac{1}{2})\right)(\tilde{z}_j^H), \tag{15}$$

where Θ^H and Θ^L denote the parameter vectors for the univariate distributions $p_{\tilde{z}^H|\Theta^H}$ and $p_{\tilde{z}^L|\Theta^L}$.

Experimental Results

The CLIC training set with images of at least 256 pixels in height or width (1732 images in total) were used for training the proposed model. Random crops of size $h = w = 256$ were extracted from the images for training. We set $\alpha = 0.5$ so that 50% of the latent representations is assigned to the LR part with half spatial resolution. Sample HR and LR latent representations are shown in Figure 3.

Considering four layers of strided convolutions (with stride of 2) and the output channel size $M = 192$ in the core encoder (Figure 2), the HR and LR latents y^H and y^L will respectively be of size 16×16×96 and 8×8×96 for training. As discussed in (Ballé et al. 2018), the optimal number of filters (i.e., N) increases with the R-D balance factor λ, which indicates that higher network capacity is required for models with higher bit rates. As a result, in order to avoid λ-dependent performance saturation and to boost the network capacity, we set $M = N = 256$ for higher bit rates (BPPs > 0.5). All models in our framework were jointly trained for 200 epochs with mini-batch stochastic gradient descent and a batch size of 8. The Adam solver with learning rate of 0.00005 was fixed for the first 100 epochs, and was gradually decreased to zero for the next 100 epochs.

We compare the performance of the proposed scheme with standard codecs including JPEG, JPEG2000 (Christopoulos, Skodras, and Ebrahimi 2000), WebP (Google Inc. 2016), BPG (both YUV4:2:0 and YUV4:4:4 formats) (Bellard 2017), the VVC Test Model or VTM 5.2 (both YUV4:2:0 and YUV4:4:4 formats) (Fraunhofer 2019), and also state-of-the-art learned image compression methods in (Minnen, Ballé, and Toderici 2018; Li et al. 2019; Lee, Cho, and Beack 2018; Zhou et al. 2019). We use both PSNR and MS-SSIM$_{\text{dB}}$ as the evaluation metrics, where MS-SSIM$_{\text{dB}}$ represents MS-SSIM scores in dB defined as: MS-SSIM$_{\text{dB}} = -10log_{10}(1 - \text{MS-SSIM})$.

The comparison results on the popular Kodak image set (averaged over 24 test images) are shown in Figure 4. For the PSNR results, we optimized the model for the MSE loss as the distortion metric d in Equation 12, while the perceptual MS-SSIM metric was used for the MS-SSIM results reported in Figure 4. In order to obtain the six different bit rates on the R-D curve illustrated in Figure 4, six models with seven different values for λ were trained.

As shown in Figure 4, our method outperforms the standard codecs such as BPG and VTM (4:2:0) as well as the state-of-the-art learning-based image compression methods in terms of both PSNR and MS-SSIM. Our method achieves ≈0.25dB lower PSNR than VTM (4:4:4). However, compared to VTM (4:2:0), the proposed approach provides ≈0.12dB better PSNR at lower bit rates (bpp < 0.5) and ≈0.5-1dB better PSNR at higher rates.

One visual example from the Kodak image set is given in Figure 5 in which our results are qualitatively compared with JPEG2000 and BPG (4:4:4 chroma format) at 0.15bpp. As seen in the example, our method provides the highest visual quality compared to the others. JPEG2000 has poor performance due to the ringing artifacts. The BPG result is smoother compared to JPEG2000, but the details and fine structures are not preserved in many areas, for example, the patterns of red feathers on the right bird's chest.

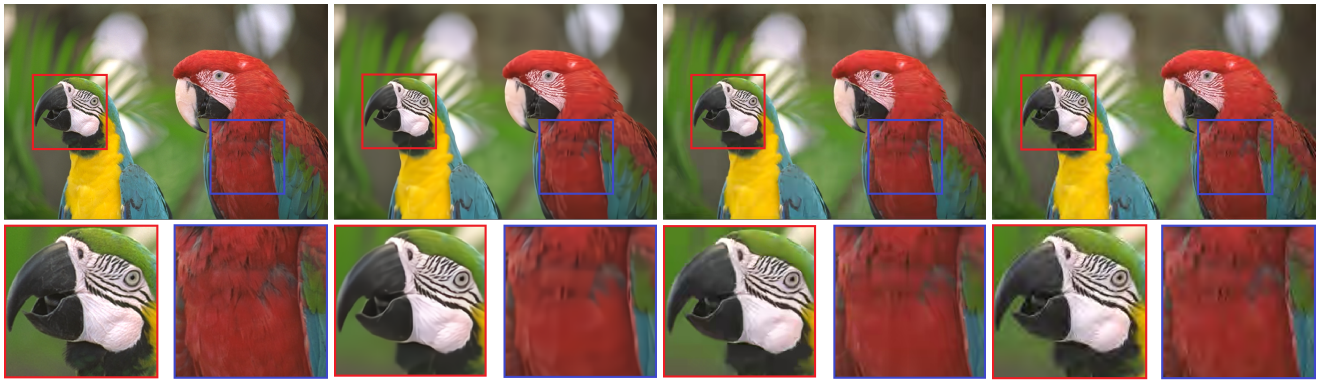
Ablation Study

In order to evaluate the performance of different components of the proposed framework, we perform some ablation studies reported in Table 1. The reported PSNR, MS-SSIM, and Inference Time are averaged over Kodak image set. The Inference Time includes the entire encoding and decoding time. All the models have been optimized for MSE distortion metric (for one single bit-rate), test with both PSNR and MS-SSIM metric.

LR Ratio: in order to study varying choices of the ratio of channels allocated to LR maps, we tested our model with three different ratios $\alpha \in \{0.25, 0.5, 0.75\}$. As summarized in Table 1, compressing 50% of the LR part to half the resolution (i.e., $\alpha = 0.5$) results in the best R-D performance at 0.345bpp (where the contributions of HR and LR latents are 0.276bpp and 0.069bpp). As the ratio decreases to $\alpha = 0.25$, less compression with a higher bit rate of 0.445bpp (0.410bpp for HR and 0.035 for LR) is obtained, while no significant gain in the reconstruction quality is achieved. Although increasing the ratio to 75% provides a better compression with 0.309bpp (high: 0.132bpp, low: 0.176bpp), it significantly results in a lower PSNR. As indicated by the number of floating point operations per second (FLOPs) in the table, larger ratio results in a faster model since less operations are required for calculating the LR maps with half spatial resolution.

Position of activation layers: in this scenario (denoted by ActOut), as in the original octave convolution (Chen et al. 2019), we apply GDN to the output HR and LR maps in GoConv, and IGDN before the input HR and LR maps for GoTConv. As the results indicate, the proposed scheme with internal activations ($\boldsymbol{\alpha}$ = 0.5) provides a better performance (with ≈0.27dB higher PNSR) since all internal feature maps corresponding to the inter- and intra- communications are benefited from the activation function.

Octave-based core auto-encoder: the proposed bi-resolution model utilizes GoConv/GoTConv units for both latents and hyper latents. In order to study the effectiveness of bi-resolution modelling of hyper latents, we also report



(a) Original (b) Ours (0.149bpp, 35.30dB, 15.25dB) (c) BPG (0.151bpp, 33.96dB, 14.44dB) (d) J2K (0.151bpp, 32.59dB, 13.43dB)

Figure 5. Kodak visual example (bits-per-pixel, PSNR, MS-SSIM$_{\text{dB}}$).

	α = 0.25	α = 0.5	α = 0.75	ActOut	CoreOct	OrgOct
BPP (HR / LR)	0.445 (0.410 / 0.035)	0.345 (0.276 / 0.069)	0.309 (0.132 / 0.176)	0.346	0.339	0.338
PSNR (dB)	32.38	32.35	31.35	32.08	31.86	28.92
MS-SSIM (dB)	15.26	15.25	15.08	14.99	14.34	12.37
FLOPs (G)	16.57	13.69	11.04	12.98	11.51	8.84

Table 1. Ablation study of different components in the proposed framework. BPP: bits-per-pixel (HR/LR: BPPs for HR and LR latents). ActOut: activation layers moved out of GoConv/GoTConv; CoreOct: proposed GoConv/GoTConv only used for the core auto-encoder; OrgOct: GoConv/GoTConv replaced by original octave convolutions.

the results in which GoConv/GoTConv are only used for the core auto-encoder latents (denoted by CoreOct). To deal with the HR/LR latents resulted from the bi-resolution core auto-encoder, we used two separate networks (similar to (Minnen, Ballé, and Toderici 2018) with Vanilla convolutions) for each of the hyper encoder, and hyper decoder. A PSNR gain of ≈0.49dB is achieved when both core and hyper auto-encoders benefit from the proposed bi-resolution model.

Original octave convolutions: in this experiment, the performance of the proposed GoConv/GoTConv architectures compared with the original octave convolutions (denoted by OrgOct) is analyzed. We replace all GoConv layers in the proposed framework (Figure 2) by original octave convolutions. For the octave transposed-convolution used in the core and hyper decoders, we reverse the octave convolution operation formulated as follows:

$$\tilde{X}^H = g(\tilde{Y}^H; \Psi^{H\to H}) + upsample(g(\tilde{Y}^L; \Psi^{L\to H}), 2),$$
$$\tilde{X}^L = g(\tilde{Y}^L; \Psi^{L\to L}) + g(downsample(\tilde{Y}^H, 2); \Psi^{H\to L}), \quad (16)$$

where $\{\tilde{Y}^H, \tilde{Y}^L\}$ and $\{\tilde{X}^H, \tilde{X}^L\}$ are the input and output feature maps, and g is vanilla transposed-convolution. Similar to the octave convolution defined in (Chen et al. 2019), average pooling and nearest interpolation are respectively used for down- and up-sampling operations. As reported in Table 1, OrgOct provides a significantly lower performance than the architecture with the proposed GoConv/GoTConv, which is due to the fixed sub-sampling operations incorporated for its inter-resolution components. The PSNR and MS-SSIM of the proposed architecture are respectively ≈3.43dB and ≈2.88dB higher than Org-Conv at the same bit rate. Compared to the other models, OrgOct has the lowest complexity with respect to FLOPs. Note that the ratio $\alpha = 0.5$ was used for the ActOut, CoreOct, and OrgOct models.

Conclusion

In this paper, we propose a new learned bi-resolution image compression and entropy model with octave convolutions in which the latents are factorized into HR and LR components, and the LR is stored at lower resolution to reduce the spatial redundancy. To preserve the spatial structure of the input, novel generalized octave convolution and transposed-convolution architectures denoted by GoConv/GoTConv are introduced. Our experiments show that the proposed method significantly improves the R-D performance and achieves the new state-of-the-art learned image compression, which even outperforms VTM (4:2:0) in PSNR.

Our method bridges the wavelet transform and deep learning-based image compression, and allows many techniques in the wavelet transform research to be applied to learned image compression. This will lead to many other research topics in the future. We also show the benefit of the proposed GoConv/GoTConv in other CNN-based computer vision applications, particularly auto-encoder-based schemes such as image denoising and semantic segmentation.

Acknowledgments

This work is supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada under grant RGPIN-2015-06522.

References

Agustsson, E.; Mentzer, F.; Tschannen, M.; Cavigelli, L.; Timofte, R.; Benini, L.; and Gool, L. V. 2017. Soft-to-Hard Vector Quantization for End-to-End Learning Compressible Representations. *arXiv preprint arXiv:1704.00648* .

Akbari, M.; Liang, J.; and Han, J. 2019. DSSLIC: Deep semantic segmentation-based layered image compression. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2042–2046.

Akbari, M.; Liang, J.; Han, J.; and Tu, C. 2020. Learned Variable-Rate Image Compression With Residual Divisive Normalization. In *IEEE International Conference on Multimedia and Expo*, 1–6.

Antonini, M.; Barlaud, M.; Mathieu, P.; and Daubechies, I. 1992. Image coding using wavelet transform. *IEEE Transactions on Image Processing* 1(2): 205–220.

Ballé, J.; Laparra, V.; and Simoncelli, E. P. 2016. End-to-end optimization of nonlinear transform codes for perceptual quality. In *Picture Coding Symposium*, 1–5.

Ballé, J.; Minnen, D.; Singh, S.; Hwang, S. J.; and Johnston, N. 2018. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436* .

Bellard, F. 2017. BPG image format. http://bellard.org/bpg. Last accessed on 2021-03-01.

Chen, Y.; Fan, H.; Xu, B.; Yan, Z.; Kalantidis, Y.; Rohrbach, M.; Yan, S.; and Feng, J. 2019. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In *IEEE International Conference on Computer Vision*, 3435–3444.

Christopoulos, C.; Skodras, A.; and Ebrahimi, T. 2000. The JPEG2000 still image coding system: an overview. *IEEE Transactions on Consumer Electronics* 46(4): 1103–1127.

Fraunhofer, H. 2019. VVC Official Test Model VTM. https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM. Last accessed on 2021-03-01.

Google Inc. 2016. WebP. https://developers.google.com/speed/webp. Last accessed on 2021-03-01.

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

Lee, J.; Cho, S.; and Beack, S. 2018. Context-adaptive entropy model for end-to-end optimized image compression. *arXiv preprint arXiv:1809.10452* .

Li, B.; Akbari, M.; Liang, J.; and Wang, Y. 2020. Deep Learning-based Image Compression with Trellis Coded Quantization. In *Data Compression Conference*, 13–22.

Li, M.; Zuo, W.; Gu, S.; You, J.; and Zhang, D. 2019. Learning Content-Weighted Deep Image Compression. *arXiv preprint arXiv:1904.00664* .

Minnen, D.; Ballé, J.; and Toderici, G. D. 2018. Joint autoregressive and hierarchical priors for learned image compression. In *Advances in Neural Information Processing Systems*, 10771–10780.

Rippel, O.; and Bourdev, L. 2017. Real-time adaptive image compression. In *International Conference on Machine Learning*, volume 70, 2922–2930.

Springenberg, J. T.; Dosovitskiy, A.; Brox, T.; and Riedmiller, M. 2014. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806* .

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

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

Vaidyanathan, P. P. 2006. *Multirate systems and filter banks*. Pearson Education India.

Van den Oord, A.; Kalchbrenner, N.; Espeholt, L.; Vinyals, O.; Graves, A.; et al. 2016. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, 4790–4798.

Zhou, J.; Wen, S.; Nakagawa, A.; Kazui, K.; and Tan, Z. 2019. Multi-scale and context-adaptive entropy model for image compression. *arXiv preprint arXiv:1910.07844* .