Two-Stage Octave Residual Network for End-to-End Image Compression

Fangdong Chen[1*], **Yumeng Xu**[1], **Li Wang**

Hikvision Research Institute
{chenfangdong, xuyumeng, wangli7}@hikvision.com

Abstract

Octave Convolution (OctConv) is a generic convolutional unit that has already achieved good performances in many computer vision tasks. Recent studies also have shown the potential of applying the OctConv in end-to-end image compression. However, considering the characteristic of image compression task, current works of OctConv may limit the performance of the image compression network due to the loss of spatial information caused by the sampling operations of inter-frequency communication. Besides, the correlation between multi-frequency latents produced by OctConv is not utilized in current architectures. In this paper, to address these problems, we propose a novel Two-stage Octave Residual (ToRes) block which strips the sampling operation from OctConv to strengthen the capability of preserving useful information. Moreover, to capture the redundancy between the multi-frequency latents, a context transfer module is designed. The results show that both ToRes block and the incorporation of context transfer module help to improve the Rate-Distortion performance, and the combination of these two strategies makes our model achieve the state-of-the-art performance and outperform the latest compression standard Versatile Video Coding (VVC) in terms of both PSNR and MS-SSIM.

Introduction

Image compression is essential for image transmission and storage, and has been studied for a long period. During the passing decades, image coding standard, including JPEG (Wallace 1992), JPEG 2000 (Rabbani and Joshi 2002), HEVC/H.265 (Sullivan et al. 2012) and VVC/H.266 (Bross et al. 2020) have been persistently developed to pursue a better Rate-Distortion performance. However, the compression rate is still proceeding slowly, and even the latest coding standard is unable to meet the fast-growing image traffic. Traditional compression codec is composed of several independently optimized modules, namely, prediction, transform, quantization, entropy coding, and loop filters. Intuitively, it is promising to achieve higher performance if the codec can be optimized as a whole. Since neural network can adjust each module automatically under a unity of purpose, the end-to-end neural network based scheme has been a popular and effective way to compress images.

The most common end-to-end image compression network base on Convolutional Neural Network (CNN) is consist of a nonlinear analysis transformation, a uniform quantizer, and a nonlinear synthesis transformation (Ballé, Laparra, and Simoncelli 2016). To pursue less coding bit rate and higher quality of reconstructed images, lots of methods have been raised and incorporated into the encoder-decoder network in recent years. For reducing the coding rate of bit stream, hyperprior model (Ballé et al. 2018) based on variational autoencoder is introduced as a powerful entropy model on the local scale parameters of the latent representation. 2D and 3D context model (Minnen, Ballé, and Toderici 2018; Lee, Cho, and Beack 2018) are developed to reduce the redundancy of latents. Probabilistic generative models (Ballé et al. 2018) are proposed to parameterize the distribution of latents for the arithmetic encoding/decoding. Attention module (Li et al. 2018; Liu et al. 2019) is introduced to this network for adapting bit allocation. In addition, for enhancing the quality of reconstructed images, ResNet (He et al. 2016) is incorporated into this network for preventing network degeneration. Based on these studies, the end-to-end image compression has already outperformed most classical standard codecs.

In the field of image compression, it is found that features with high frequency differ from those with low frequency in coding characteristics (Devore, Jawerth, and Lucier 1992). High frequency features refer to the areas that gray value changes rapidly, while low frequency features refer to the areas that gray value changes smoothly. However, most end-to-end methods omit the differences of multi-frequency, and simply mix and process the different features using a set of unified kernels. In recent years, octave convolution (OctConv) (Chen et al. 2019), which has achieved great success in many computer vision tasks (Fan et al. 2019; Xu et al. 2020), shows great potential for solving this problem. It is proposed to factorize feature maps into high and low frequency groups (with different resolutions of feature maps), and process them separately with different kernels. Moreover, a modified OctConv called generalized octave convolution (GoConv) (Akbari et al. 2020) is developed and applied in the image compression model (Ballé et al. 2018), which shows processing low- and high-frequency features sepa-

[1] Equal contribution, [*] corresponding author.

rately has distinct advantages in image compression task.

In the image compression task, the network should down sampling the feature maps and reduce the redundancy of latent representations to save the bit rate of the stream, meanwhile the network also has to store useful information as much as possible in the latent representations to reconstruct the high quality images in the decoder. However, there are two factors that may limit the performance of OctConv and GoConv under this characteristic for image compression. On one hand, both OctConv and GoConv mix the down-sampling operation and inter-frequency communication operation in one stage, thus the inter frequencies communicate in an information-loss stage, which is inefficient. On the other hand, the correlation information between high and low frequency latents is not utilized for arithmetic coding in the architectures, which may limit the redundancy reduction in the latent coding part. The detailed analysis of these factors will be shown in the Section Problem Definition.

To address these problems above, we propose a Two-stage Octave Residual block (ToRes block), which strips the sampling operation from OctConv to solve the inter-frequency issue and to strengthen the capability of preserving useful information. Besides, a context transfer module is designed to entirely use the correlation information between high and low frequency latents.

The contributions of this paper are summarized as follows:

- We dig deeper into the advantages and limitations of OctConv in image compression task and design the ToRes block which strips the sampling operation from OctConv to strengthen the capability of preserving useful information in the end-to-end image compression architecture. The proposed module ToRes block can utilize the advantages of OctConv and avoid the limitations in image compression.
- We propose a context transfer module. It can utilize the correlation between high and low frequency latents to reduce redundancy, which is significantly complementary to the joint entropy model (Minnen, Ballé, and Toderici 2018).
- Extensive experiments demonstrate that our approach achieves the state-of-the-art (SOTA) performance in terms of both PSNR and MS-SSIM metrics (Wang, Simoncelli, and Bovik 2003) on common benchmarks. The approach outperforms VVC (VTM10.0) (JVET 2020) with as high as 0.5 dB in terms of the PSNR, which is significant improvement in visual data compression field.

Related Work

Learning-based Image Compression

In recent years, a majority of deep learning-based image compression models have been developed based on Convolutional Neural Networks (CNN) (Ballé, Laparra, and Simoncelli 2016; Ballé et al. 2018; Minnen, Ballé, and Toderici 2018) and Recurrent Neural Networks (RNN) (Toderici et al. 2016; Minnen et al. 2018; Lin et al. 2020). In terms of CNN, the well-known encoder-decoder architecture is developed by Ballé (Ballé, Laparra, and Simoncelli 2016). This architecture is composed of an analysis transform encoder, a uniform quantizer and a synthesis transform decoder. In this study, novel activation functions called generalized divisive normalization (GDN) and inverse GDN are introduced. GDN/IGDN is inspired by models of neurons in biological visual systems, and has proven effective in Gaussianizing image densities. On the basis of this work, in order to capture spatial redundancy in the latent representation produced by the encoder, a powerful entropy model, hyperprior, based on variational autoencoders (VAE) is proposed (Ballé et al. 2018). This model allows evaluating the standard deviations of latent representation and calculating their distribution by Gaussian Scale Model (GSM). To exploit probabilistic structure in the latents and make further improvement on bit reduction, the autoregressive model of the latent representations is proposed and integrated with hyperprior as a hierarchical entropy model (Minnen, Ballé, and Toderici 2018). The hierarchical entropy model can reduce spatial redundancy using not only side information but also neighboring elements in latents. This study is the first learned method that outperforms BPG.

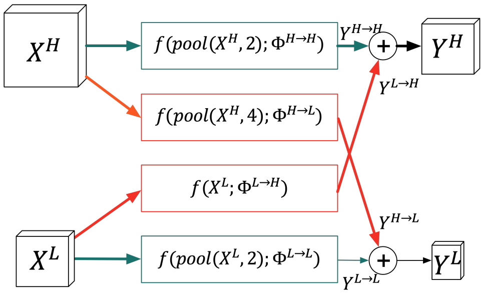

X^H
$f(pool(X^H, 2); \Phi^{H \to H})$
$Y^{H \to H}$
Y^H
$f(pool(X^H, 4); \Phi^{H \to L})$
$Y^{L \to H}$
$f(X^L; \Phi^{L \to H})$
$Y^{H \to L}$
X^L
$f(pool(X^L, 2); \Phi^{L \to L})$
$Y^{L \to L}$
Y^L

Figure 1: The architecture of the down-sampling OctConv convolutions.

As a widely applied technique in natural language processing tasks, the attention mechanism is also incorporated in the field of image compression. Liu *et al.* (Liu et al. 2019) introduce non-local attention into the VAE structure. It can capture both local and global correlations and generate attention masks for adapting bit allocation to reduce the rate of less important pixels. In addition to these methods, parameterized entropy models, such as Gaussian Mixture Model (GMM) and discretized Gaussian Mixture Likelihoods (DGML), have also been studied and applied in the field of image compression. Since single Gaussian Scale Model (GSM) cannot achieve arbitrary likelihoods (Cheng et al. 2020), DGML which is an accurate and flexible entropy model, is proposed for compressing the remaining redundancy.

Octave Convolution

In general, natural images contain two kinds of information, namely high-frequency information and low-frequency information. High-frequency information refers to the information that changes rapidly, such as the detailed information like boundaries. Low-frequency information refers to the information that changes smoothly, such as the background

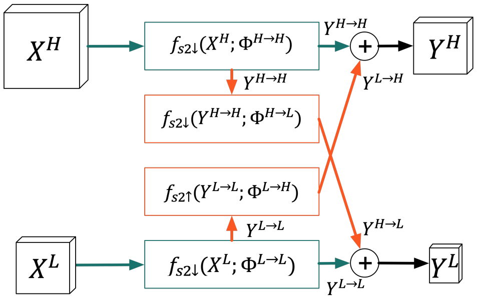

X^H
$f_{s2\downarrow}(X^H;\Phi^{H\to H})$
$Y^{H\to H}$
Y^H
$Y^{H\to H}$
$Y^{L\to H}$
$f_{s2\downarrow}(Y^{H\to H};\Phi^{H\to L})$
$f_{s2\uparrow}(Y^{L\to L};\Phi^{L\to H})$
$Y^{L\to L}$
$Y^{H\to L}$
X^L
$f_{s2\downarrow}(X^L;\Phi^{L\to L})$
$Y^{L\to L}$
Y^L

Figure 2: The architecture of the down-sampling GoConv convolution.

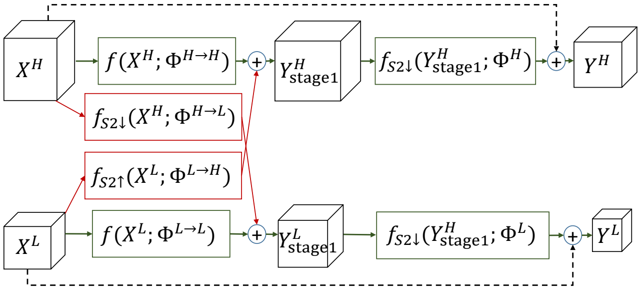

X^H
$f(X^H;\Phi^{H\to H})$
Y^H_{stage1}
$f_{S2\downarrow}(Y^H_{\text{stage1}};\Phi^H)$
Y^H
$f_{S2\downarrow}(X^H;\Phi^{H\to L})$
$f_{S2\uparrow}(X^L;\Phi^{L\to H})$
X^L
$f(X^L;\Phi^{L\to L})$
Y^L_{stage1}
$f_{S2\downarrow}(Y^H_{\text{stage1}};\Phi^L)$
Y^L

Figure 3: Detailed architectures of the down-sampling ToRes block. The resolutions of output tensors (Y^H, Y^L) are reduced to half of the resolutions of their corresponding input tensors (X^H, X^L). Dotted arrows refer to shortcut connections which are sampling strided convolutions.

in images. In a recent study (Chen et al. 2019), OctConv is proposed to processe high-frequency and low-frequency features separately. This proposed multi-frequency feature representation method stores and processes low-frequency features by mapping them to low-resolution tensors for reducing spatial redundancy.

The architectures of the OctConv are shown as Figure 1, where X^H and X^L denote the high-frequency and low-frequency input tensors, respectively. $Y^{H\to H} = f(X^H;\Phi^{H\to H})$ and $Y^{L\to L} = f(X^L;\Phi^{L\to L})$ denote the output tensors of intra-frequency update, and $Y^{H\to L} = f(pool(X^H,2);\Phi^{H\to L})$ and $Y^{L\to H} = upsample(f(X^L;\Phi^{L\to H}),2)$ denote the output tensors of inter-frequency communication. $pool(X,n)$ denotes average pooling with stride n. It is worth noting that unlike the traditional method to separate different frequencies, the high- and low-frequency feature maps refer to the feature maps with different resolutions. With the intra-frequency update and inter-frequency communication, the OctConv network learns to separate two kinds of features into two groups of feature maps by itself. The convolutional kernel Φ is split into four groups: $\Phi^{H\to H}$, $\Phi^{H\to L}$, $\Phi^{L\to H}$ and $\Phi^{L\to L}$, for processing different input tensors to different output tensors .

Since low-frequency feature maps are processed lower resolution, OctConv can significantly save the usage of computational resources, such as memory. This operation is also reported to improve the performance of many computer vision tasks by replacing vanilla convolution. On image segmentation task, Fan *et al.*(Fan et al. 2019) build an accurate retinal vessel segmentation neural network using OctConv and achieves comparable performance to other state-of-the-art methods with a faster processing speed. On image classification task, Xu *et al.*(Xu et al. 2020) propose a multiscale octave 3D CNN for hyperspectral image classification which outperforms the state-of-the-art deep learning methods.

Problem Definition

The success of OctConv on image processing tasks indicates that it also has the potential for contributing to image compression field, because of the ability to reduce the spatial redundancy of low-frequency feature maps and processing different features separately. Due to the plug-and-play character, it is easy to incorporate OctConv into the architecture of image compression as mentioned above.

However, there are two reasons that OctConv is unsuitable for replacing the down and up sampling convolutions directly. In image compression task, keeping more meaningful information in the limited bit stream is essential for reconstructing high quality images. The first reason is that all down sampling operations are achieved by average pooling in OctConv, which may not preserve enough useful spatial information of the input. The second reason is that as shown in Figure 1, when OctConv does down-sampling operation to reduce the resolution of output feature maps to the half of that of input feature maps, the stride length of inter-frequency pooling has to be 4, which is too long to keep enough spatial information and may lead to an astounding increase of distortion.

Therefore, Akbari *et al.*(Akbari et al. 2020) propose a generalized octave convolution (GoConv), as shown in Figure 2. The adjustments are in two main aspects. Firstly, the pooling operation in OctConv is replaced by the strided convolutions for preserving and reconstructing more information. Secondly, the output tensors of intra-frequency convolutions $(Y^{H\to H}, Y^{L\to L})$ are regarded as the input tensors of inter-frequency convolutions for making a shorter length of down-sampling stride.

Problems occur when GoConv is incorporated into the image compression architecture. For convenience, down-sampling GoConv with stride of 2 is taken as an example. As shown in Figure 1, in OctConv, the output of intra-frequency convolutions $(Y^{H\to H}, Y^{L\to L})$ only contribute to the corresponding frequency features, which means that neurons of these two convolutions only need to learn information of one frequency features as much as possible. However, in GoConv shown in Figure 2, the output of intra-frequency convolutions $(Y^{H\to H}, Y^{L\to L})$ are also the input of the inter-frequency convolutions, which leads to a dilemma that neurons for intra-frequency convolutions need to learn both high-frequency information and low-frequency information. This issue may make it challenging to train kernel parameters and decrease the performance of the model. In addition, when it comes to the low-to-high convolution in GoConv, a down-sampling operation $f_{s2\downarrow}(\cdot)$ is done on the low frequency feature maps, and then followed by an up-sampling operation $f_{s2\uparrow}(\cdot)$. This down-and-up operation is redundant and may increase the distortion loss. Therefore, conducting the down-sampling operation and inter-frequency communication operation in one stage is not effective for image compression.

Proposed Method

ToRes Block

To address the aforementioned problems, we propose a novel convolutional unit called Two-stage Octave Residual block (ToRes block), which combines OctConv with the structure of ResNet and strips the sampling operation from inter-frequency communication operation with two stages. As shown in Figure 3, ToRes block is composed of an OctConv and two vanilla convolutions for two frequencies. Down and up sampling operations are conducted by these two strided vanilla convolutions instead of OctConv.

A down-sampling ToRes block is formulated as follows:

$$Y^H_{stage1} = f^{H\rightarrow H}(X^H;\Phi^{H\rightarrow H}) + f^{L\rightarrow H}_{(s2\uparrow)}(X^L;\Phi^{L\rightarrow H}) \tag{1}$$

$$Y^L_{stage1} = f^{L\rightarrow L}(X^L;\Phi^{L\rightarrow L}) + f^{H\rightarrow L}_{(s2\downarrow)}(X^H;\Phi^{H\rightarrow L}) \tag{2}$$

$$Y^H_{stage2} = f^H_{(s2\downarrow)}(Y^H_{stage1};\Phi^H) \tag{3}$$

$$Y^L_{stage2} = f^L_{(s2\downarrow)}(Y^L_{stage1};\Phi^L) \tag{4}$$

$$Y^H = Y^H_{stage2} + f_{shortcut}(X^H) \tag{5}$$

$$Y^L = Y^L_{stage2} + f_{shortcut}(X^L) \tag{6}$$

where $Y^H_{stage1}, Y^L_{stage1}$ denote the output tensors of OctConv in the first stage, and $Y^H_{stage2}, Y^L_{stage2}$ denote the output tensors of vanilla convolutions in the second stage. $f(\cdot;\Phi)$ denotes a convolution operation with parameter Φ, and $s2\downarrow$ and $s2\uparrow$ denote the down and up sampling operations with stride 2 correspondingly. $f_{shortcut}(\cdot)$ denotes the skip connection in ResNet, which is a down-sampling strided convolution. With the analysis, it can be seen that the first stage of the ToRes block can focus on the inter-frequency communication with as much information as possible, and no unnecessary down-sampling operation is introduced in this stage. The down-sampling operation for certain frequency is later conducted in the second stage to save coding bits.

In this ToRes block, all inter-frequency sampling operations are operated by strided convolutions, which can keep more information and achieve better performance than average pooling. Moreover, referred to Eq.(3) and (4), strided vanilla convolutions are used to conduct the sampling operations instead of OctConv in the whole convolutional unit aspect, which means the resolution of output feature maps of ToRes block is half of the resolution of input. Similarly, the up-sampling ToRes block has the same structure but replaces the down-sampling vanilla convolutions and shortcut with up-sampling ones, respectively.

This structure can merge the advantages of OctConv into the image compression architecture as well as avoid the drawbacks analyzed in Section Problem Definition. Furthermore, the structure of the residual block is adopted for improving the rate-distortion performance and preventing network degeneration.

Context Transfer Module

To compress the information in latent representations and reduce the coding bit rate, the hierarchical entropy model called joint entropy model (Minnen, Ballé, and Toderici 2018) is proposed to estimate the entropy parameters for arithmetic coding. Since in our proposed ToRes network, the latents are separated into high- and low-frequency groups, thus there are still correlation between these two latents, which would be redundancy that joint entropy model is unable to capture. Therefore, we propose a context transfer module for further reducing the redundancy between these two latents. In the encoding/decoding procedure, high-frequency latents are first encoded/decoded. Then the high-frequency latents are processed by our context transfer module to extract useful information, which is used for estimation of low-frequency latents entropy parameters and reducing the bit rate of low-frequency bit stream.

The context transfer module is composed of a down-sampling strided convolution and two residual blocks. It allows reducing the resolution of high-frequency latent representation to the same resolution of low-frequency latents. The location of context transfer module is shown in Figure 4. The input of the context transfer model is the quantized high-frequency latents, and the output is concatenated with low-frequency outputs of context model and hyperprior, and then the concatenated tensors are adopted as the input of entropy parameter layers.

Network Structure

As shown in Figure 4, the proposed network architecture adopts an improved VAE network as the backbone. The overall network can be divided into core network (left part) and sub-network (right part) according to their functions.

The core network aims to generate latent representations of input images, and then reconstruct images based on these latents. The core network is mainly formed by two parts: an analysis transform encoder and a synthesis transform decoder. Then ToRes blocks are adopted in this encoder-decoder architecture for improving the performance of the compression model.

The sub-network aims to estimate the parameters of probabilistic models over quantized latent representations for arithmetic coding and decoding. It contains one hyperprior (hyper-analysis transform and hyper-synthesis transform), two context models (for two frequency latents), one context transfer module and entropy parameter layers. The convolution layers in the hyperprior are also replaced by OctConv for processing multi-frequency latents together. In context models, 5x5 masked convolution is implemented for capturing the correlation with the neighboring elements.

Moreover, DGML (Cheng et al. 2020) is adopted as the probabilistic model. Parameters of the probabilistic model are estimated by the output of the hyperprior, the context model and the context transfer module. The data from these

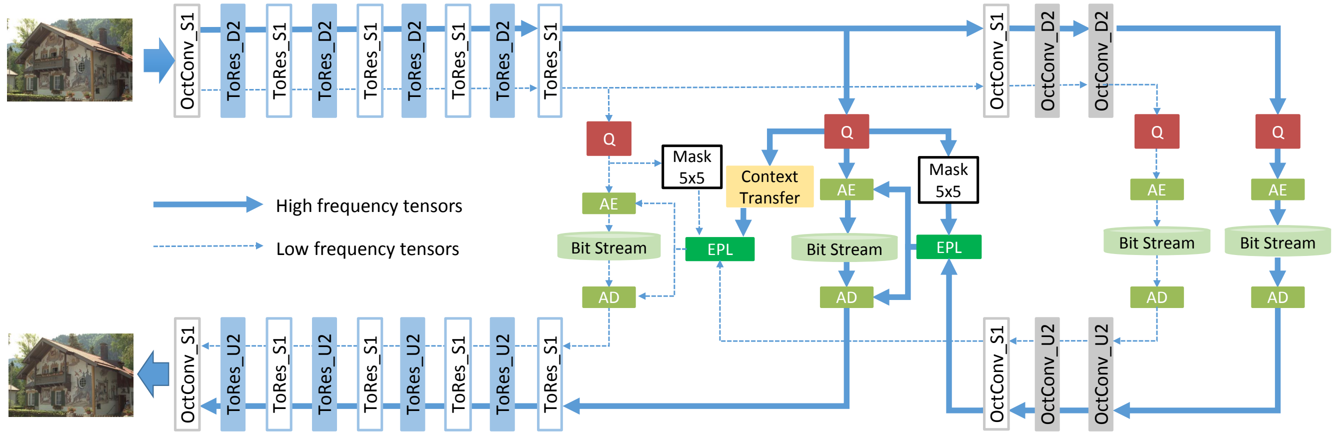


Figure 4: The network architecture of our proposed model. The suffix D2 and U2 denote down and up sampling operations with stride of 2, respectively. And the suffix S1 represents no resolution change. The analysis transform encoder (top left parts) contains an input OctConv which divides the input images into multi-frequency feature maps, four down-sampling ToRes blocks, and four original ToRes blocks which do not operate sampling. The synthesis transform decoder (bottom left parts) is symmetric. It contains four up-sampling ToRes blocks, four original ToRes blocks, and an output OctConv which combines multi-frequency feature maps together and builds the output images. There are three 5x5 OctConv layers in hyper-analysis transform and hyper-synthesis transform, respectively. Q represents the additive uniform noise for training, or uniform quantizer for the test. AE and AD denote arithmetic encoder and decoder, respectively. EPL represents entropy parameter layers.

three parts is concatenated as the input of entropy parameter layers for calculating K groups of mean, scale and weight of Gaussian entropy models. In our network, the value of K is set to 3 for the entropy model.

Experiments

Implementation Details

We filter the images in ImageNet database (Deng et al. 2009) by size from 500 × 500 to 1000 × 1000, and crop them randomly to the size of 256 × 256 for training our model. The training batch size is set to 8. Each model is trained up to 10^6 iterations for each λ to achieve a stable performance. During training, the models are optimized using Adam optimizer (Kingma and Ba 2014). The learning rate is set to 10^{-4} for the first 900k iterations, and reduced to 10^{-5} for the last 100k iterations. The models are optimized for mean square error (MSE) or MS-SSIM metrics.

The process of image compression can be formulated as follows:

$$y_h, y_l = g_a(x; \phi) \tag{7}$$

$$\hat{y_h}, \hat{y_l} = Q(y_h, y_l) \tag{8}$$

$$\hat{x} = g_s(\hat{y_h}, \hat{y_l}; \theta) \tag{9}$$

where x and $\hat{x}$ denote input raw images and output reconstructed images. y_h, y_l, $\hat{y_h}$ and $\hat{y_l}$ denote high and low frequency latent representations before and after quantization $Q(\cdot)$. $g_a(\cdot; \phi)$ and $g_s(\cdot; \theta)$ refer to the analysis transform and the synthesis transform with parameters ϕ and θ. Then the loss function are shown as follow:

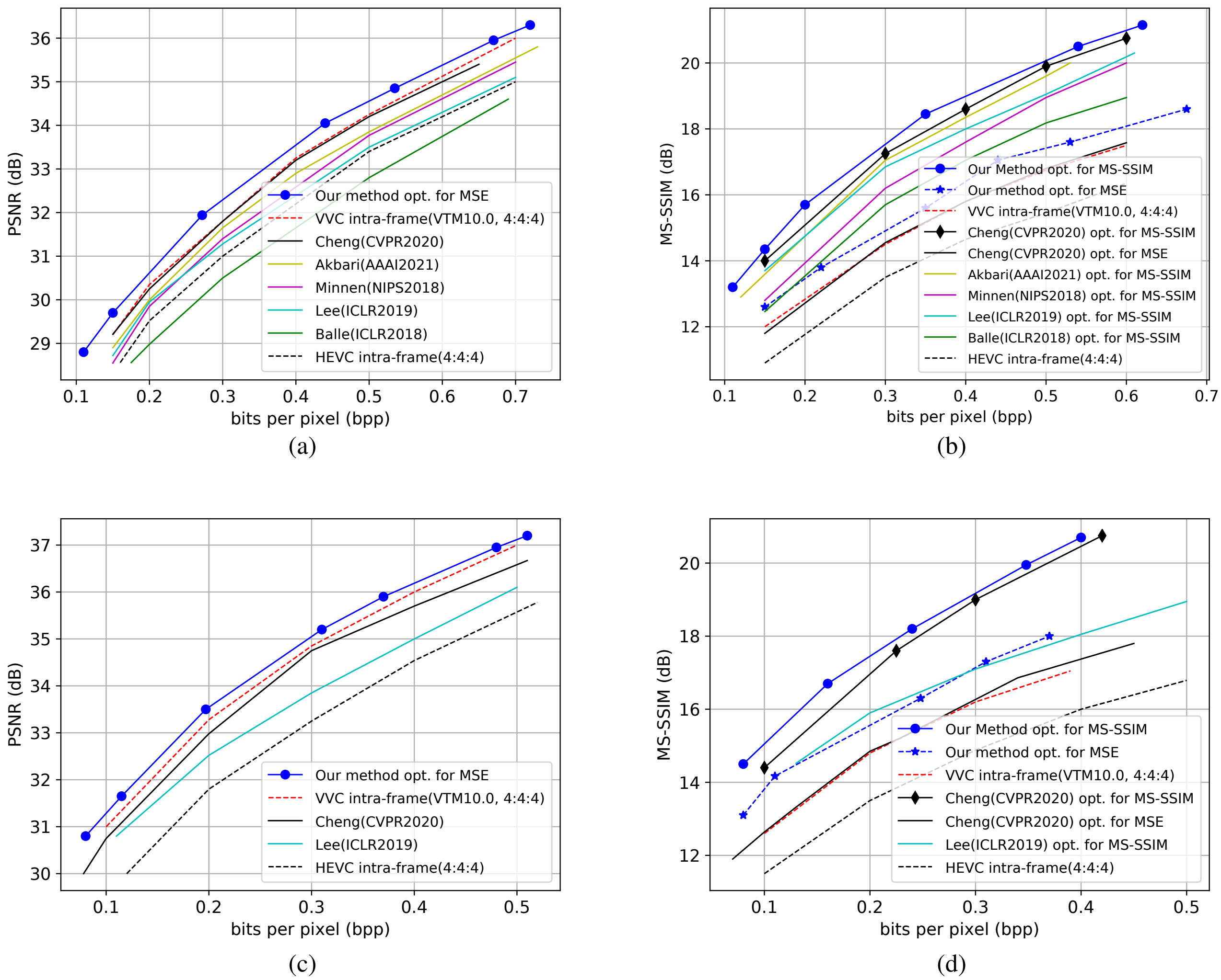
$$\begin{aligned} L =& \lambda \cdot D(x, \hat{x}) + R_{y_h} + R_{y_l} + R_{z_h} + R_{z_l} \\ =& \lambda \cdot D(x, \hat{x}) + \mathbb{E}[-log_2 p_{\hat{y}_h}(\hat{y}_h)] + \mathbb{E}[-log_2 p_{\hat{y}_l}(\hat{y}_l)] \\ &+ \mathbb{E}[-log_2 p_{\hat{z}_h}(\hat{z}_h)] + \mathbb{E}[-log_2 p_{\hat{z}_l}(\hat{z}_l)] \end{aligned} \tag{10}$$

where λ is the Lagrange multiplier to control the rate-distortion tradeoff. $D(\cdot)$ refers to the distortion between original input image x and reconstructed output image $\hat{x}$. R_{y_h}, R_{y_l}, R_{z_h} and R_{z_l} represent the estimated bit rates of high-frequency and low-frequency latent representation and side information of hyperprior, respectively. When optimized for MSE, λ belongs to the set $\{0.002, 0.003, 0.007, 0.015, 0.02, 0.03, 0.035\}$. When optimized for MS-SSIM, distortion is defined by $D(\cdot) = 100 \times (1-\text{MS-SSIM}(x, \hat{x}))$, and λ belongs to the set $\{0.025, 0.04, 0.07, 0.2, 0.4, 0.5\}$.

Other parameters of our proposed models are given in Table 1. Ratio of low-frequency feature maps is set to $\alpha_{in} = \alpha_{out} = \alpha$ throughout the network, except the input OctConv and the output OctConv where $\alpha_{in} = 0$, $\alpha_{out} = \alpha$ and $\alpha_{in} = \alpha$, $\alpha_{out} = 0$. Since the models at higher bit rates need more high-frequency information and higher network capacity, the ratio of low-frequency feature maps α is changed with λ.

Evaluation

For evaluating the model, Kodak lossless image database (Franzen 1999) with 24 uncompressed 768 x 512 images and CVPR2018 Workshop and Challenge on Learned Image Compression (CLIC) validation dataset P (professional) (CLIC 2018) with 41 high resolution images are used. Rate-Distortion is used to evaluate the model at different bit rate.



(a)

(b)

(c)

(d)

Figure 5: Rate-distortion curves on Kodak dataset and CLIC dataset. (a) PSNR (dB) on Kodak dataset. (b) MS-SSIM (dB) on Kodak dataset. (c) PSNR (dB) on CLIC dataset. (d) MS-SSIM (dB) on CLIC dataset.

λ	α	N	M
0.003	0.75	192	192
0.005	0.5	256	256
0.01	0.25	256	256
0.02	0.25	320	320

Table 1: Parameters of our proposed models. M denotes the numbers of the output convolutions of analysis transform encoder and hypersynthesis transform, and N denotes the numbers of channels of all other convolutions in our architecture.

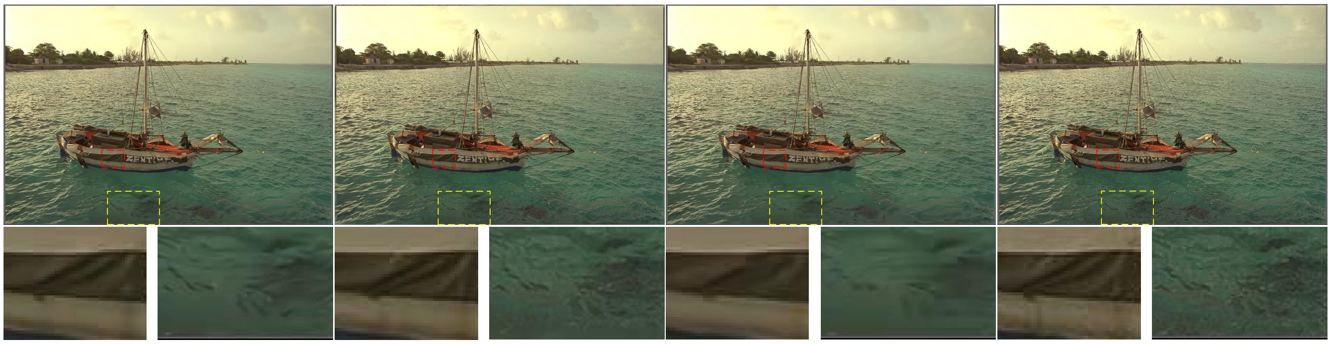
Distortion loss is measured by either PSNR or MS-SSIM between original input images and output reconstructed images, corresponding to marks “opt. for MSE” and “opt. for MS-SSIM” (as shown in the legends of Figure 5), respectively.

Rate-Distortion Performance

we compare the rate-distortion performance of our model with some previous learning-based methods and traditional compression standards. Learning-based methods include Ballé (Ballé et al. 2018), Lee (Lee, Cho, and Beack 2018), Minnen (Minnen, Ballé, and Toderici 2018) and Cheng (Cheng et al. 2020), and traditional compression standards include HEVC intra-frame coding and VVC intra-frame coding. MS-SSIM values are converted into decibels $(-10log_{10}(1-\text{MS-SSIM}))$.

The results shown in Figure 5 demonstrate that our proposed method outperforms other learning-based works and achieves the state-of-the-art in terms of both PSNR and MS-SSIM. Specially, on the most frequently used database for image compression (Kodak database), the proposed method outperforms VVC (VTM10.0) with as high as 0.5 dB in terms of the PSNR, which is significant improvement nowadays. On CLIC dataset, the performance of our model still outperforms other learning-based work and VVC, which would help to verify the robustness of our models on high resolution images.

Figure 6 shows the subjective quality performance of our



(a) (b) (c) (d)

Figure 6: Visual examples kodim06 from Kodak dataset. (a) Ours opt. for MSE (0.4971bpp, PSNR: 33.19dB, MS-SSIM: 0.9767). (b) Ours opt. for MS-SSIM (0.5098bpp, PSNR: 29.14dB, MS-SSIM: 0.9899). (c) VTM10.0 (0.4783bpp, PSNR: 31.88dB, MS-SSIM: 0.9707). (d) Ground truth.

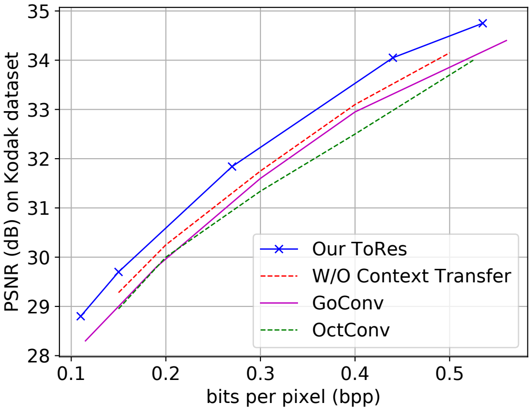


Figure 7: Ablation studies.

model and VTM10.0 on kodim 24. Compared with VTM10.0, our reconstructed images optimized by MSE and our reconstructed images optimized by MS-SSIM both show better visual quality and keep more details, such as the details of the ripple.

Ablation Studies

To demonstrate the performance of our proposed components, ablation studies are performed as follows.

ToRes block vs. OctConv and GoConv In this part, the performance of OctConv, GoConv and proposed ToRes block are compared. We train two models, the first one replaces all ToRes blocks with OctConv in our proposed network, and the second replaces all ToRes block with GoConv. The R-D curves of these two models and our proposed model are shown in Figure 7. Compared with OctConv, our proposed ToRes block achieves about 0.2∼0.5 dB PSNR increase. Moreover, compared with GoConv, ToRes block can still achieve about 0.2∼0.3 dB PSNR increase at the same bit rate. It can be observed that the ToRes block can bring significant coding gain than the previously proposed structures, which fully demonstrates the effectiveness of the ToRes block.

Context Transfer Module In order to evaluate the performance of the context transfer module, we retrain the same image compression networks without this module. As shown in Figure 7, the modified model has about 0.02∼0.05 bpp bit-rate increase (about 10%) at the same distortion without this module. This result shows that context transfer module can evidently reduce the rate of bit stream and improve the performance for our model.

Complexity

OctConv has less computational complexity than vanilla convolution, and it has the same number of parameters with vanilla convolution (Chen et al. 2019). Considering the whole model, our ToRes network has less complexity than Cheng (Cheng et al. 2020) including FLOPs and the number of parameters. In terms of FLOPs (with the input image size 1920*1080, channel number 192 for Cheng (Cheng et al. 2020) and 256 for our model), our encoder and decoder are about 1.80T (when α=0.5), while encoder and decoder models in Cheng (Cheng et al. 2020) are about 1.92T. In addition, the parameter numbers of our encoder and decoder are about 9.65M and 8.66M, respectively, while the parameter numbers of Cheng (Cheng et al. 2020) encoder and decoder are about 12.01M and 10.36M, respectively.

Conclusion

In this paper, we address the problem that both OctConv and GoConv have limitations when applied in image compression model, and propose an effective ToRes network with context transfer module. ToRes block is developed for introducing the advantages of OctConv for image compression task and avoiding its limitations by stripping the sampling operation from it with two-stage design. Context transfer module is incorporated into our network for capturing the redundancy between multi-frequency latents produced by the ToRes block. The results of experiments show that our proposed method outperforms existing learning-based methods and well-known traditional compression standards including VVC intra coding, and achieves the state-of-the-art performance in terms of both PSNR and MS-SSIM.

References

Akbari, M.; Liang, J.; Han, J.; and Tu, C. 2020. Generalized Octave Convolutions for Learned Multi-Frequency Image Compression. *arXiv preprint arXiv:2002.10032*.

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

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*.

Bross, B.; Chen, J.; Liu, S.; and Wang, Y. 2020. Versatile Video Coding (Draft 10). *Document JVET-S2001 of JVET (Jul. 2020)*.

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 *Proceedings of the IEEE International Conference on Computer Vision*, 3435–3444.

Cheng, Z.; Sun, H.; Takeuchi, M.; and Katto, J. 2020. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7939–7948.

CLIC. 2018. Workshop and Challenge on Learned Image Compression. *source: http://www.compression.cc/challenge/*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Devore, R. A.; Jawerth, B.; and Lucier, B. J. 1992. Image compression through wavelet transform coding. *IEEE Transactions on Information Theory*, 38(2): 719–746.

Fan, Z.; Mo, J.; Qiu, B.; Li, W.; Zhu, G.; Li, C.; Hu, J.; Rong, Y.; and Chen, X. 2019. Accurate retinal vessel segmentation via octave convolution neural network. *arXiv preprint arXiv:1906.12193*.

Franzen, R. 1999. Kodak lossless true color image suite. *source: http://r0k.us/graphics/kodak*, 4(2).

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

JVET, T. 2020. VTM software repository, version VTM-10.0. https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM.

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

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

Li, M.; Zuo, W.; Gu, S.; Zhao, D.; and Zhang, D. 2018. Learning convolutional networks for content-weighted image compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3214–3223.

Lin, C.; Yao, J.; Chen, F.; and Wang, L. 2020. A Spatial RNN Codec for End-to-End Image Compression. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13269–13277.

Liu, H.; Chen, T.; Guo, P.; Shen, Q.; Cao, X.; Wang, Y.; and Ma, Z. 2019. Non-local attention optimized deep image compression. *arXiv preprint arXiv:1904.09757*.

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.

Minnen, D.; Toderici, G.; Covell, M.; Chinen, T.; Johnston, N.; Shor, J.; Hwang, S. J.; Vincent, D.; and Singh, S. 2018. Spatially adaptive image compression using a tiled deep network. *arXiv preprint arXiv:1802.02629*.

Rabbani, M.; and Joshi, R. 2002. An overview of the JPEG 2000 still image compression standard. *Signal processing: Image communication*, 17(1): 3–48.

Sullivan, G. J.; Ohm, J.-R.; Han, W.-J.; and Wiegand, T. 2012. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12): 1649–1668.

Toderici, G.; Vincent, D.; Johnston, N.; Hwang, S. J.; Minnen, D.; Shor, J.; and Covell, M. 2016. Full Resolution Image Compression with Recurrent Neural Networks. *arXiv preprint arXiv:1608.05148*.

Wallace, G. K. 1992. The JPEG still picture compression standard. *IEEE transactions on consumer electronics*, 38(1): xviii–xxxiv.

Wang, Z.; Simoncelli, E. P.; and Bovik, A. C. 2003. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, 1398–1402. Ieee.

Xu, Q.; Xiao, Y.; Wang, D.; and Luo, B. 2020. Csamso3dcnn: Multiscale octave 3d cnn with channel and spatial attention for hyperspectral image classification. *Remote Sensing*, 12(1): 188.