Practical Stacked Non-local Attention Modules for Image Compression

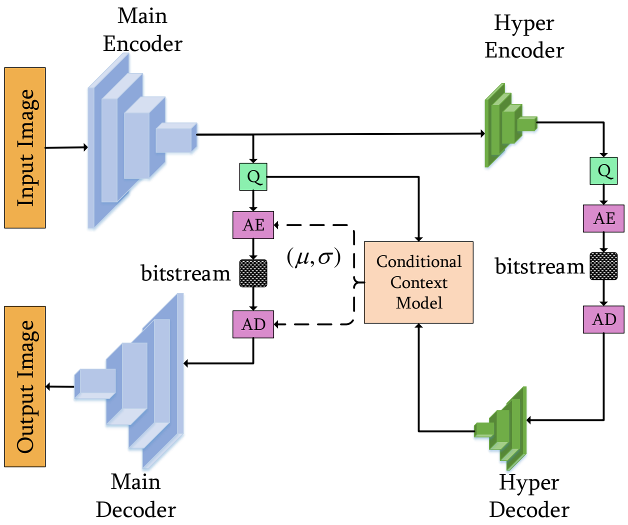
Haojie Liu* Tong Chen* Qiu Shen† Zhan Ma†
Vision Lab, Nanjing University, Nanjing, China
*{haojie, tong}@smail.nju.edu.cn †{shenqiu, mazhan}@nju.edu.cn

Abstract

In this paper, we proposed a stacked non-local attention based variational autoencoder (VAE) for learned image compression. We use a non-local module to capture global correlations effectively that can't be offered by traditional convolutional neural networks (CNNs). Meanwhile, layer-wise self-attention mechanisms are widely used to activate/preserve important and challenging regions. We jointly take the hyperpriors and autoregressive priors for conditional probability estimation. For practical application, we have implemented a sparse non-local processing via maxpooling to greatly reduce the memory consumption, and masked 3D convolutions to support parallel processing for autoregressive priors based probability prediction. A post-processing network is then concatenated and trained with decoder jointly for quality enhancement. We have evaluated our model using public CLIC2019 validation and test dataset, offering averaged 0.9753 and 0.9733 respectively when evaluated using multi-scale structural similarity (MS-SSIM) with bit rate less than 0.15 bits per pixel (bpp).

1. Introduction

Learned image compression methods [3, 6, 9] are usually based on the well-known autoencoders, transforming raw image data into compressible latent features by stacked CNNs. These methods achieve joint rate-distortion optimization (RDO) in an end-to-end learning manner. A variety of loss functions can be adapted for learning towards individual optimization requirements, for example, mean-squared-error (MSE) loss for pixel distortion measurement, MS-SSIM loss for structure similarity, etc. Thanks to the advances in high-efficiency transforms (e.g., generalized divisive normalization [2]), differentiable quantization (e.g., derivable rounding [10], adding uniform noise [2] and soft-to-hard decision [1]), and learned entropy model (e.g., masked convolutions [6], hyperpriors for probability estimation [3] and joint priors from autoregressive neighbors and hyperpriors [7]), learned image compression methods present a great success compared with traditional codec such as JPEG, JPEG2000, BPG both subjectively and objectively.



Parameter Details

Main Encoder	Conv5x5 c32	ReLU	Conv5x5 c64s2	NLAM c192 s8	Conv5x5 c192s2	NLAM c192 s4	Conv5x5 c192s2	NLAM c192 s2	Conv5x5 c192s2	Conv3x3 c64
Hyper Encoder	NLAM c192s2	Conv5x5 c192s2	NLAM c192s2	Conv5x5 c192s2	NLAM c192s2	Conv3x3 c42				
Context Model	MaskedConv 5x5x5 c24	Conv1x1x1 c48	LeakyReLU	Conv1x1x1 c96	LeakyReLU	Conv1x1x1 c2				

Figure 1. Illustration of our image compression framework using stacked non-local attention module (NLAM). Q denotes the quantization. AE and AD represent arithmetic encoding and decoding. Network parameters are shown below. For example, "Conv5×5c64s2" illustrates a convolution layer of 5×5 kernel size, 64 channels and 2-pixel stride. In NLAM, "NLAMc192s8" represents that all the convolutions have identical 192 channels and the sparse sampling factor of maxpooling is 8 for memory reduction (e.g., 1/64).

VAE architecture was first proposed in [3] to use hyperpriors for better entropy modeling of quantized latent features. In the meantime, Generalized divisive normalization (GDN) was widely applied because of its reported high-efficiency than other activiations such as ReLU, leakyReLU etc. Later, explicit content weighted importance maps were developed to guide adaptive bit allocation for better quality at the same bit rate in [6] and [4]. In contrast to widely used MSE loss, Rippel *et al.* [9] attempted to apply MS-SSIM loss function, resulting in much better visual quality (particularly for low bit rate scenario). Meanwhile, multiscale discriminators can be also included to do adversarial optimization, offering visually appealing reconstructions even with bit rate < 0.1 bpp.

However, all aforementioned methods are constrained to the local operations due to limited receptive field of CNNs. They could not capture long-range dependencies effectively and caused optimization problems even with deeper architecture. Furthermore, existing adaptive bits allocation methods always depend on an integral importance map at the bottleneck layer which is only applied to the quantized latent features. The importance map generation method is also too simple to handle complicate content.

In this paper, we applied non-local modules (NLM) [11] in compression framework enabling the model to capture global correlations, and combined it with stacked CNNs to further generate advanced attention mask for layer-wise feature activation. Traditional non-local operations require a large amount of memory, making it impractical in applications. Thus, we introduced the sparse non-local processing via maxpooling to greatly reduce the memory consumption but preserve the efficiency. Meanwhile, we employed a masked 3D convolutions to support parallel prediction of probaility estimation using autoregressive neighbors, leading to considerable decoding time reduction according to our extensive simulations. Here we only use 1/W decoding time compared with original masked convolutions. Another post-processing network is provided and trained with decoder jointly to improve the image reconstruction quality.

2. Non-local Attention Modules (NLAM) for Image Compression

Figure 1 depicts the proposed image compression framework, consisting of several piped NLAM for both main and hyper autoencoders. The main autoencoder is used to obtain the reconstructed image and the hyper autoencoder generates quantized features at much lower resolution for conditional probability modeling of the quantized latent features from main autoencoder. We have applied a 3D conditional context model to joint leverage the autoregressive and hyper priors for better entropy modeling.

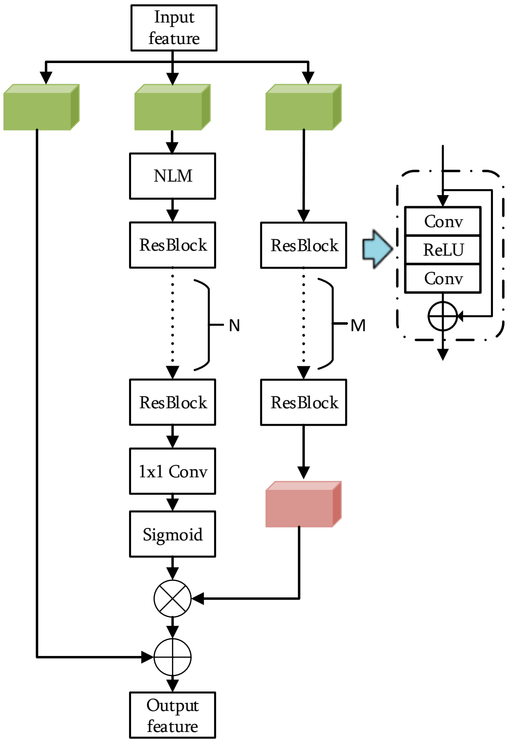


Figure 2. Diagram of NLAM embedded in proposed image compression framework. $N = M = 4$ in this work.

2.1. Sparse NLAM

Inspired by recent works in [11], we proposed to use NLAMs to guide adaptive feature generation. As shown in Fig. 2, the basic NLAM has two branches. The main branch uses M residual blocks to extract the main features. The mask branch utilizes a non-local module (NLM) followed by N residual blocks, and finally generates an implicit attention mask by cascading a 1×1 convolution and a sigmoid function. Attention mask commonly has its variable ranging from 0 to 1 continuously which can be used to give efficient weights for features extracted from the main branch. In addition, residual connections is used for better convergence. Here, we remove the batch normalization layers (BN) in the residual blocks and do not add non-linear activation after residual connection.

In reality, NLM requires a large amount of memory to host a correlation matrix at size of $HW \times HW$. Note that H and W are the height and width for input feature map. We then apply the maxpooling to enable the sparse NLAM by downscaling the correlation matrix layer by layer. We set scaling factor s to balance overall NLAM under limited memory. Note that we use Gaussian embedding in our framework to achieve non-local operations [11].

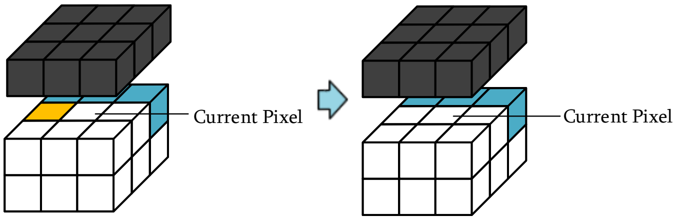


Figure 3. Illustration of the parallel 3D masked convolutions. *Left*: the masked convolution takes all the former pixels into consideration following strictly sequential processing pixel by pixel; *Right*: Parallel processing at each line by removing left neighbor for convolution, leading to noticeable speed up of probability prediction.

2.2. Parallel 3D Masked Convolution

PixelCNNs and PixelRNNs [8] are usually proposed to capture the natural image probability distribution in a specific prediction order. Both 2D and 3D masked convolutions can be extended to model the entropy rate in the quantized latent features pixel by pixel. Although the masked convolutions can leverage the neighbor pixels to predict the current pixels efficiently, it usually leads to a great computational penalty because of the strictly pixel-by-pixel processing, making the compression framework far from the practical application.

As shown in Fig. 3, we take a 3×3×3 convolution for example. As we used the left neighbor (highlighted in Yellow) for masked convolutions, each current pixel is predicted in a raster scan manner, and it takes $H \times W \times C$ convolutions to complete all feature maps. Here, C represents the number of channels of the quantized latent features. The right modified masked convolution is used in our context model which only needs $H \times C$ convolutions by removing the left neighor. This ensures the parallel processing for each line. Simulations show that negligible performance impact is reported.

2.3. Entropy Modeling

We build different density models for $\hat{y}$ and $\hat{z}$. Here, $\hat{y}$ and $\hat{z}$ represent the quantized latent features and the hyper encoded features respectively. For $\hat{z}$, we model the priors using a non-parametric, fully factorized density model following [3]. We convolve it with a standard uniform density to get $p_{\hat{z}|\psi}$,

$$p_{\hat{z}|\psi}(\hat{z}|\psi) = \prod_i (p_{z_i|\psi^{(i)}}(\psi^{(i)}) * \mathcal{U}(-\frac{1}{2}, \frac{1}{2}))(\hat{z}_i), \qquad (1)$$

where $\psi^{(i)}$ represents the parameters of each univariate distribution $p_{\hat{z}|\psi^{(i)}}$.

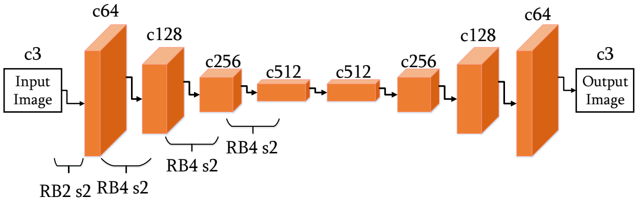


Figure 4. The post-processing network used in this work. For example,”RB2s2” represents 2 residual blocks with a 2×2 downsampling convolutional layer.

For $\hat{y}$, each element $\hat{y}_i$ can be modeled as a Gaussian distribution by joint autoregressive priors and hyperpriors,

$$p_{\hat{y}}(\hat{y}_i|\hat{y}_1, ..., \hat{y}_{i-1}, \hat{z}) = \prod_i (\mathcal{N}(\mu_i, {\sigma_i}^2) * \mathcal{U}(-\frac{1}{2}, \frac{1}{2}))(\hat{y}_i), \qquad (2)$$

where $\hat{y}_1, \hat{y}_2, ..., \hat{y}_{i-1}$ denote the causal (and possibly reconstructed) pixels and $\hat{y}_i$ is the current pixel to be predicted, its μ_i and σ_i are predicted by the joint priors. Then we can simply use the cumulative distribution function (CDF) to calculate the probability of each symbol by Gaussian distribution.

We evaluate the bits of $\hat{y}$ and $\hat{z}$ using:

$$R_{\hat{y}} = -\sum_i \log_2(p_{\hat{y}}(\hat{y}_i|\hat{y}_1, ..., \hat{y}_{i-1}, \hat{z})), \qquad (3)$$

$$R_{\hat{z}} = -\sum_i \log_2(p_{\hat{z}_i|\psi^{(i)}}(\hat{z}_i|\psi^{(i)})). \qquad (4)$$

2.4. Post-processing Network for Quality Enhancement

Post-processing is widely used in tradtional codec to further enhance the quality of reconstructed image. Several methods utilize CNN networks to learn a non-linear mapping to remove the block artifacts caused by lossy compression. We introduce one in this work that is trained with the decoder jointly, as shown in Fig. 4. In the other words, the post-processing network is cascaded with the decoder network which makes the network deeper and achieve better reconstructions.

2.5. Model Adaptation-Based Rate Control

Three different models (level i ∈1,2,3) are used for RDO to meet bit rate budget imposed by the CLIC test.. First, all the images are encoded at the highest bpp (i = 3) as the initial state. Then, images with the minimum $\frac{\text{MS}-\text{SSIM}_i - \text{MS}-\text{SSIM}_{i-1}}{\text{filesize}_i - \text{filesize}_{i-1}}$ will be encoded at lower bpp by adapting models iteratively until the overall file size meets the requirement (lower than 0.15 bpp).

3. Experimental Discussion

We use **COCO** [5] training dataset to train our framework. We randomly resize the images and take 256 × 256

Table 1. Result on CLIC2019 validation dataset

Entry	MS-SSIM	PSNR	Image Size
TucodecSSIM	0.9758	29.84	4692810
NJUVisionSSIMF	0.9753	29.61	4715606
ETRI	0.9751	29.70	4722275
JointSSIM	0.9751	29.76	4721983
....	....	....	....

cropped patches for preprocessing. We choose MS-SSIM as our distortion metric and the loss function is

$$L = \lambda(1 - \mathrm{MS} - \mathrm{SSIM}) + R_y + R_z, \qquad (5)$$

where we set different λs to achieve rate-distortion trade-off to generate several models for variable compression ratio. In our experiment, we set λ to 4, 8, 12 to obtain our models. We replaced the MSE with MS-SSIM [12] because it is reported to have better correlations with our human visual perceptual sensation, especially at low bitrate. All of the modules in our framework are trained together. Batch size is set to 64 and finally trained on 4-GPUs in parallel. The learning rate is first set to 10^{-4} and is halved every 5 epoches until reaching the convergence. After that, we utilize the post-processing network to enhance the images, resulting in another 0.0004 to 0.0008 MS-SSIM improvement.

We have evaluated our model on CLIC2019 validation and test dateset, and achieved averaged 0.9753 and 0.9733 MS-SSIM respectively with bit rate < 0.15 bpp. As shown in Table 1, we achieved the second place on the CLIC validation leadboard among all teams participating the MS-SSIM distortion evaluation category.

4. Conclusion

We proposed a practical stacked non-local attention based variational autoencoder for learned image compression and achieved noticeable performance efficiency on public CLIC test datasets. Sparse sampling for correlation matrix is introduced to greatly reduce the memory consumption in non-local modules. By remove left neighbor for prediction, we offer parallel 3D masked convolutions for probability estimation using autoregressive priors, leading to considerable decoding time speed-up (i.e., from pixel-by-pixel processing to line-by-line processing when using parallel pixel prediction). We think the non-local attention modules is crucial for the improvement in our compression framework. Certainly, a deeper context model will further improve our coding efficiency but require more decoding time. Block-based optimization used in traditional codec can enforce more parallelism but it might also introduce blocky or boundary artifacts.

References

[1] Eirikur Agustsson, Fabian Mentzer, Michael Tschannen, Lukas Cavigelli, Radu Timofte, Luca Benini, and Luc V Gool. Soft-to-hard vector quantization for end-to-end learning compressible representations. In *Advances in Neural Information Processing Systems*, pages 1141–1151, 2017. 1

[2] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016. 1

[3] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018. 1, 3

[4] Mu Li, Wangmeng Zuo, Shuhang Gu, Debin Zhao, and David Zhang. Learning convolutional networks for content-weighted image compression. *arXiv preprint arXiv:1703.10553*, 2017. 2

[5] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 3

[6] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. Conditional probability models for deep image compression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 3, 2018. 1, 2

[7] David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. In *Advances in Neural Information Processing Systems*, pages 10794–10803, 2018. 1

[8] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016. 3

[9] Oren Rippel and Lubomir Bourdev. Real-time adaptive image compression. *arXiv preprint arXiv:1705.05823*, 2017. 1, 2

[10] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. *arXiv preprint arXiv:1703.00395*, 2017. 1

[11] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018. 2

[12] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003. 4