Self-Supervised Variable Rate Image Compression using Visual Attention

Abhishek Kumar Sinha, S. Manthira Moorthi, Debajyoti Dhar
Signal and Image Processing Group
Space Applications Centre, Ahmedabad
{aks, smmoorthi, deb}@sac.isro.gov.in

Abstract

The recent success of self-supervised learning relies on its ability to learn the representations from self-defined pseudo-labels that are applied to several downstream tasks. Motivated by this ability, we present a deep image compression technique, which learns the lossy reconstruction of raw images from the self-supervised learned representation of SimCLR ResNet-50 architecture. Our framework uses a feature pyramid to achieve the variable rate compression of the image using a self-attention map for the optimal allocation of bits. The paper provides an overview to observe the effects of contrastive self-supervised representations and the self-attention map on the distortion and perceptual quality of the reconstructed image. The experiments are performed on a different class of images to show that the proposed method outperforms the other variable rate deep compression models without compromising the perceptual quality of the images.

1. Introduction

Image compression is fundamental to the intersection of computer vision, signal processing, and information theory. The constant evolution in the image compression methods is critical to meet the ever-growing demand for data transfer and storage of data. The past few decades have witnessed the development of various traditional and neural codecs for efficient compression. The deep neural codecs use convolutional neural networks or generative models to learn the compressible latent representation of the image. The variable rate models offer the flexibility to adjust the compression ratio using a single trained architecture. Toderici *et al*. [17] proposed the first end-to-end network to achieve variable rate compression using LSTM. Balle *et al*. [5] introduced a trainable decorrelation non-linear normalization technique, called Generalized Divisive Normalization (GDN), to save more bits and achieve better compression. The GDN is used as the activation in all the image compression models for higher compression. Toderici *et al*. [19] introduced RNN based full resolution image compression method, which outperformed the traditional codecs. Balle *et al*. [3] uses a fully factorized prior model for bit rate estimation in the end-to-end trainable network. It uses a non-parametric piecewise linear density model to learn each factor of factorized prior. Islam *et al*. [9] utilizes the quantization step to control the variable bit rate. The method involves the RNN based quantization with the shallow synthesis and analysis architectures. The overall performance is improved by using an LSTM network to reduce unnecessary information. Johnston *et al*. [10] uses the recurrent network based convolutional architecture with spatially adaptive rates. The proposed architecture improves spatial diffusion to effectively capture and propagate the information through the hidden states. All of these methods deploy high capacity networks to achieve superior performance and are prone to training data overfitting. These shortcomings can be overcome if the condensed and meaningful representations of the images, that capture variety of features, are used as inputs to the model.

Self-supervised representation learning is already being used for various downstream tasks and therefore learning efficient representations in pretext tasks becomes crucial to improve the performance of the downstream tasks. Chen *et al*. [6, 7] proposed SimCLR for self-supervised learning using contrastive loss framework. SimCLR learns representations by maximizing agreement between differently augmented views of the same example using a contrastive loss over latent representation.

The ability of self-supervised models to learn and capture the variations from tremendously diverse samples is key to reducing the training complexity of the downstream models. With the benefit of learning downstream tasks in a self-supervised setting, we propose a self-supervised variable rate deep image compression technique. The major contributions of the paper are as follows:

1. The paper discusses feature pyramid based network in the form of encoder-decoder setup to achieve variable rate compression.

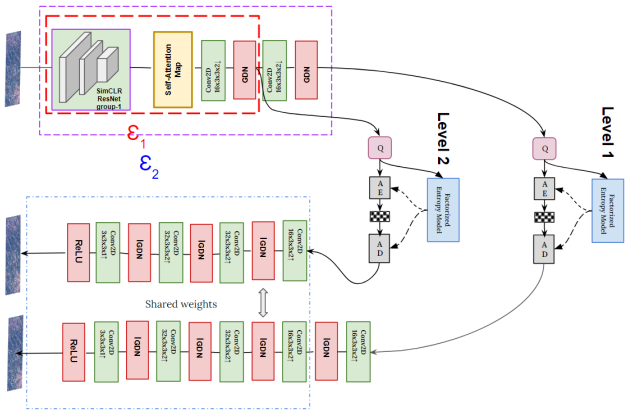


Figure 1. Illustration of self-supervised attention guided image compression system. In this figure, the encoder takes the SimCLR ResNet-50's $group-1$ features as the input. AE and AD stand for Arithmetic Encoding and Arithmetic Decoding respectively. Though this architecture depicts two levels of feature coding, it can be extended to more numbers of levels.

2. The network is trained as a downstream task to learn the compressible features from the pretrained SimCLR's representations. The ablation study further highlights the gain in the rate-distortion curve on using learned representations.

3. The model attends the salient region using a self attention map over the SimCLR's representations to filter out the least important features. Furthermore, the placement of self-attention network is also studied to improve the compression efficiency.

2. Proposed Methodology

2.1. Architecture

Fig. 1 describes the complete architecture of the proposed method. The model primarily consists of four modules: Encoder $\mathcal{E}$, Quantizer $\mathcal{Q}$, Entropy coder $\mathcal{H}$, and Decoder $\mathcal{D}$. Given an image $x \in \mathbb{R}^{H\times W\times C}$, the pre-trained SimCLR ResNet-50 in the encoder generates the features ϕ. Following [23], these features are being attended by a self-attention map $\mathcal{A} = \sigma(\phi)$ to distinguish the distortion prone regions based on the energy score. These feature maps are further used to generate a multi-scale feature pyramid $f_i(\mathcal{A})$ for variable bit rates. The sequence of layers and modules responsible for generating a particular bit rate is referred to as level. The encoder for i^{th} level is denoted by $\mathcal{E}_i = f_i(\mathcal{A}(\phi))$. These features are quantized, entropy coded $H(\mathcal{Q}(\mathcal{E}_i))$. The decoder uses a singleton architecture to decompress the pyramid of features. The decoder $\mathcal{D}_i(\theta_{k:1\leq k\leq i})$ at i^{th} level shares the weights $\theta_{1\leq k\leq i-1}$ with the decoder network at $(i-1)^{th}$ level.

2.1.1 Feature Pyramid Encoder

Instead of naively learning the features from the input image, the encoder uses the learned representations of the SimCLR model to compress the image as a downstream task. We extract group 1 ResNet-50 features for the compression. Because of the bit allocation constraints, the representation may not be efficient enough for compression. So they are further refined by a self-attention map. The self-attention module [23] computes energy score using a learned key and query values. The energy score is normalized using the softmax function to generate the corresponding attention coefficients. The self-attention map gives more weight to the regions, which impacts the perceived quality of the reconstructed image. The resultant features are convolved down to different scales. After each convolution layer, Generalized Divisive Normalization (GDN) [5] is applied to decorrelate the features before entropy coding.

2.1.2 Quantization and Entropy coding

Recent studies [3, 10] have introduced different stochastic perturbations to model the quantization in training. The quantization step is modeled as the additive uniform noise for non-zero gradient during back propagation. The quantized value of $\mathbf{z}$ is computed as $\hat{\mathbf{z}} = \mathcal{Q}(\mathbf{z}) = \mathbf{z} + U\big(-\frac{1}{2}, \frac{1}{2}\big)$

The probability distribution p($\mathbf{z}$) of the quantized latent representation $\hat{z}$ is modeled using non-parametric fully factorized model given by,

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

Following [4], the $p_{\mathbf{z}}$ is modeled using its cumulative $c : \mathbb{R} \rightarrow [0, 1]$. The vector ψ^i represents the parameters of univariate distribution $P_{z_i|\psi^i}$.

2.1.3 Decoder

The decoder is framed in such a way that it requires only one architecture to decode the variable rate compressed data. For simplicity, the decoder $\mathcal{D}$ can be seen as a group of hierarchical networks with shared weights where each network $\mathcal{D}_i$ learns the additional set of trainable parameters θ_i along with the parameters of $\mathcal{D}_{i-1}(\theta_{k:1\leq k\leq k-1})$ to decode a variable rate data $\hat{x}_i$. The architecture in Fig. 1 includes two levels of hierarchical decoding and therefore is capable of decoding the image at two bit rates. The number of levels in the feature pyramid can be increased further by adding subsequent convolutional layers in the encoder.

2.2. Rate-Perception-Distortion Trade-off

The loss function jointly optimizes three parameters: bit rate, distortion and perceptual loss. The perceptual loss $\mathcal{P}$ is computed as the Mean Squared Error (MSE) between the

features of pre-trained VGG-19 architecture, and the distortion $d(.,.)$ is computed using MSE between ground truth and reconstructed image. The overall loss function is given as,

$$\mathcal{L}(\theta,\phi) = \sum_i \alpha_i \mathbb{E}[d(\mathbf{x}, \mathcal{D}_i(\mathcal{Q}(\mathcal{E}_i(\mathbf{x};\theta)); \phi_{1\leq k\leq i}))] + \sum_i \beta_i \mathbb{E}[\mathcal{P}(\mathbf{x}, \mathcal{D}_i(\mathcal{Q}(\mathcal{E}_i(\mathbf{x};\theta)); \phi_{1\leq k\leq i}))] + \sum_i \gamma_i H_i \quad (2)$$

In equation 2, the index i represents the i^{th} level in the feature pyramid, and the parameters α_i, β_i and γ_i are the Lagrangian multipliers to control the trade-off among distortion d, perceptual quality $\mathcal{P}$ and the bit rate H_i, respectively. Since the features in the i^{th} level is the downscaled version of the $(i+1)^{th}$ level, the larger compression ratio is achieved in the i^{th} level. The Lagrangian multipliers must be chosen accordingly to control the relative distortions and the perceptual qualities at different levels of the feature pyramid and should follow the relation, $\alpha_i < \alpha_{i+1}$, $\beta_i < \beta_{i+1}$, $\gamma_i > \gamma_{i+1}$.

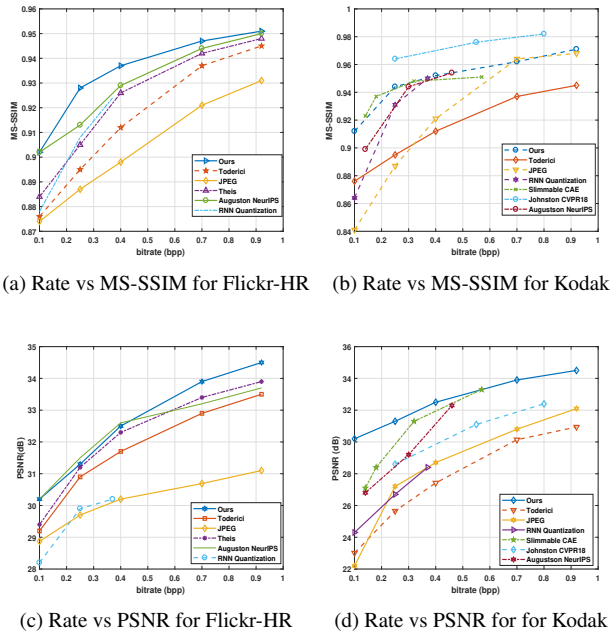
2.3. Bit Rate Inequality

We present the mathematical arguments for achieving variable bit rate in feature pyramid. For simplicity, we assume a feature pyramid with intermediate features h_1 and h_2 such that $h_2 = GDN(\mathbf{W}h_1)$. From Tishby *et al.* [16], it is known that $H(h_1) \geq H(h_2)$. The equality occurs if and only if h_1 and h_2 have injective mapping. Due to invertibility of GDN function [2] and $\mathbf{W} \neq 0$, h_1 and h_2 are one-to-one mapped in feature pyramid and can potentially lead to equal bit rates. Consider $w(t)$ as the weights causing equal entropies at t^{th} iteration. Following [8, 14], the L_2 distance of weights $w(t)$ from initialization $w(0)$ is upper bounded by radius $R = \left(\frac{n^{\frac{3}{2}}}{m^{\frac{1}{2}}\lambda_0\delta}\right)$ with probability at least 1-δ, where m,n are number of parameters in network and number of training samples respectively, and λ_0 is the smallest eigenvalue of gram matrix H^∞ given by $H^\infty_{ij} = \mathbb{E}_{u\sim\mathbb{N}(0,I)}[(h^i_1)^T h^j_1 1_{(u^T h^i_1 \geq 0, u^T h^j_1 \geq 0)}]$. To avoid the equality, the weights must stay outside of hyper-sphere of radius R. Solving for $\mathbf{W}$, the learned weights will lie out of this hyper-sphere for $\left(\frac{n^3}{\lambda_0^2\delta^2||GDN^{-1}(h_2)h_1^T(h_1h_1^T)^{-1} - w(0)||_2^2}\right)$ number of parameters in the network with probability at least 1-δ. Alternatively, a network with sufficiently large number of parameters can avoid the bit rate equality in feature pyramid.

2.4. Training

The model is trained using the CLIC 2020 dataset [18] and is evaluated on randomly selected 50 Flickr High-Resolution images and all 24 Kodak images using distortion and perceptual quality metrics. Following [4,5], The distortion is measured using MS-SSIM [21] and PSNR in RGB space. A no-reference perceptual quality measure, Perceptual Index [12, 13], is included to observe the perception-distortion trade-off.

The architecture is implemented in the Tensorflow environment and trained for the bit rates in the range of 0.1 bpp and 1.05 bpp. The architecture is optimized for the loss function described in the previous section. We used Adam optimizer [11] with the parameters $\beta_1 = 0.99$ and $\beta_2 = 0.999$ and initial learning rate of 10^{-3}, which is subsequently reduced to 5×10^{-4} after MSE loss reaches 330. The training images are cropped to 240×240 with a batch size of 16.



(a) Rate vs MS-SSIM for Flickr-HR (b) Rate vs MS-SSIM for Kodak

(c) Rate vs PSNR for Flickr-HR (d) Rate vs PSNR for for Kodak

Figure 2. Rate Distortion curves for Kodak and Flickr-HR images. The first row (a-b) illustrates the MS-SSIM variation with respect to bit rate, and the second row (c-d) demonstrates the PSNR variation.

3. Results

3.1. Qualitative and quantitative analysis

The plots in Fig. 2 quantitatively compare the performance for the bit rates in the range of 0.1 and 0.9 bpp. The rate-distortion performance is compared with JPEG [20], Islam *et al.* [9], Johnston *et al.* [10], Theis *et al.* [15], Augustson *et al.* [1], and Yang *et al.* [22]. It is observed that our model outperforms the others in terms of average PSNR and lacks behind Johnston *et al.* [10] in terms of average MS-SSIM. The JPEG performs relatively worse at lower bit rates due to the independent quantization of Discrete Cosine Transform (DCT) coefficients. Qualitative analysis is

(a) Kodim-10 (b) Ours (c) Toderici (d) JPEG

(e) Flickr2K-000224 (f) Ours (g) Toderici (h) JPEG

Figure 3. Qualitative comparison for Kodak image and Flickr HR image.

	bpp	PSNR	MS-SSIM	NIQE	PI	Ma
Kodim-10						
Ours	0.295	**30.38**	0.91	**5.22**	**3.93**	**7.35**
Tod [19]	0.297	26.24	0.89	6.07	5.97	3.32
JPEG	0.312	26.76	0.87	7.38	4.09	7.18
Johnston [10]	0.291	27.2	**0.94**	5.59	4.32	6.96
Flickr-2K 000224						
Ours	0.299	**30.62**	**0.93**	**3.29**	**2.56**	8.82
Tod [19]	0.303	26.45	0.87	4.66	3.18	8.30
JPEG	0.34	26.58	0.86	5.72	4.49	6.78
Johnston [10]	0.289	27.91	0.92	4.18	2.62	**9.16**

Table 1. Quantitative comparison of Kodak image (Fig. 3 (a-d)), and Flickr HR image (Fig. 3)(e-h). Best results are bolded.

shown in Fig. 3, and the images are compared with various models in Table 1 to show the out-performance of the proposed method in terms of perceptual quality.

4. Ablation studies

4.1. Impact of learned representations

To study the impact of SimCLR's features, we train another network, called the baseline, with similar architecture that learns the compressible features directly from the raw image. This network is trained under the same setting. Referring to the curves in Fig. 4, even the baseline outperforms the Toderici in terms of MS-SSIM and shows comparable performance in terms of PSNR. The baseline is slightly worse than the SimCLR features based model. Since the input images contain noise and implicit bias, it requires additional complexity in the network to learn good features. The SimCLR features provide efficient representations that are already compensated for the irrelevant information in the training data.

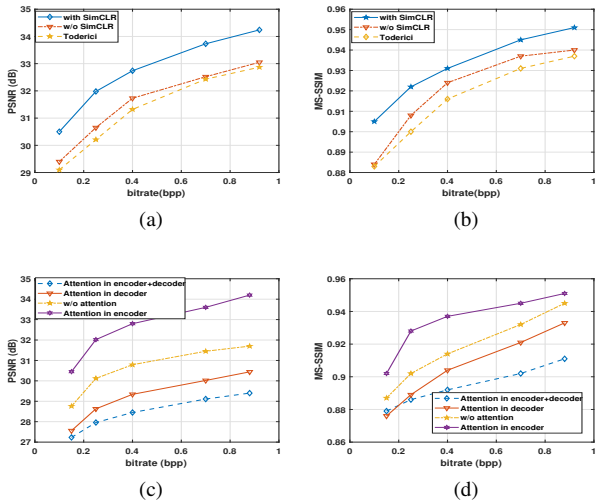


Figure 4. The curves in the top ((a)-(b)) compare the performance of the proposed method with and without SimCLR's features. The curves in the bottom ((c)-(d)) describe the impact of self-attention map and its placement in the architecture

4.2. Impact of Self-Attention map

We consider four different setups in the study, including no self-attention module in the model, self-attention module just before the last convolution layer in the decoder, self-attention module just after SimCLR in the encoder, and self-attention module in both encoder and decoder. Fig. 4 quantitatively compares the four cases using PSNR and MS-SSIM. The self-attention map in the decoder degrades the overall performance of the network. This can be attributed to the fact that the encoder extracts only relevant features for learning representations and the decoder learns to maximize the mutual information between the latent representations and the output image (Tishby *et al*. [16]). The addition of a self-attention module on the decoder's side unnecessarily scales representations and reduces the overall performance. The self-attention map just after the SimCLR features in the encoder guides bit allocation and consequently reduces the bit rate significantly.

5. Conclusion

The application of self-supervised learning reduces the complexity of the network to achieve similar performance. In addition, we provided theoretical support for the variable rate compression in the feature pyramid. We experimentally validate the use of self-attention and its position in the network to improve compression efficiency and perform thorough evaluation and comparison to the popular methods in traditional and neural image coding.

References

[1] Eirikur Agustsson, Fabian Mentzer, Michael Tschannen, Lukas Cavigelli, Radu Timofte, Luca Benini, and Luc Van Gool. Soft-to-hard vector quantization for end-to-end learning compressible representations. In *Proceedings of the Advances in Neural Information Processing Systems*, 2017. 3

[2] Johannes Balle', Valero Laparra, and Eero P. Simoncelli. Density modeling of images using generalized normalization transformation. In *Proceedings of the International Conference on Learning Representations*, 2016. 3

[3] Johannes Balle'., Valero Laparra, and Eero P. Simoncelli. End-to-end optimized image compression. In *Proceedings of the International Conference on Learning Representations*, 2017. 1, 2

[4] Johannes Balle', David Minnen, Saurabh Singh, Sun Jing Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *Proceedings of the International Conference on Learning Representations*, 2018. 2, 3

[5] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. End-to-end optimization of nonlinear transform codes for perceptual quality. In *Proceedings of the Picture Coding Symposium*, 2016. 1, 2, 3

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 1

[7] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020. 1

[8] Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *Proceedings of International Conference on Learning Representations*, 2018. 3

[9] Khawar Islam, L. Minh Dang, Sujin Lee, and Hyeonjoon Moon. Image compression with recurrent neural network and generalized divisive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1875–1879, Jun 2021. 1, 3

[10] Nick Johnston, Damien Vincent, David Minnen, Michele Covell, Saurabh Singh, Troy Chinen, Sung Jin Hwang, Joel Shor, and George Toderici. Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2, 3, 4

[11] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, 2015. 3

[12] Chao Ma, Chih-Yuan Yang, Xiaokang Yang, and Ming-Hsuan Yang. Learning a no-reference quality metric for single-image super-resolution. In *Computer Vision and Image Understanding*, 2017. 3

[13] Anish Mittal, Rajiv Soundararajan, and Alan C. Bovik. Making a completely blind image quality analyzer. *IEEE Signal processing Letters*, 22:209–212, 2013. 3

[14] Litu Rout. Why adversarial interaction creates non-homogeneous patterns: A pseudo-reaction-diffusion model for turing instability. In *AAAI Conference on Artificial Intelligence*, 2021. 3

[15] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. In *Proceedings of the International Conference on Learning Representations*, 2017. 3

[16] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5, 2015. 3, 4

[17] George Toderici, Sean M. O'Malley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell, and Rahul Sukthankar. Variable rate image compression with recurrent neural networks. In *Proceedings of the International Conference on Learning Representations*, 2016. 1

[18] G. Toderici, W. Shi, R. Timofte, L. Theis, J. Balle, E. Agustsson, N. Johnston, and F. Mentzer. Workshop and challenge on learned image compression (clic2020), 2020. 3

[19] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 4

[20] Gregory K Wallace. The jpeg still picture compression standard. *In IEEE transactions on consumer electronics*, 38:xviii–xxxiv, 1992. 3

[21] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers*, volume 2, pages 1398–1402 Vol.2, 2003. 3

[22] Fei Yang, Luis Herranz, Yongmei Cheng, and Mikhail GMozerov. Slimmable compressive autoencoders for practical neural image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4998–5007, June 2021. 3

[23] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018. 2