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 employ 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.
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 $E$, Quantizer $Q$, Entropy coder $H$, and Decoder $D$. Given an image $x \in \mathbb{R}^{H \times W \times C}$, the pre-trained SimCLR ResNet-50 in the encoder generates the features $\phi$. Following [23], these features are being attended by a self-attention map $A = \sigma(\phi)$ to distinguish the distortion prone regions based on the energy score. These features are further used to generate a multi-scale feature pyramid $f_i(A)$ 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 $E_i = f_i(A(\phi))$. These features are quantized, entropy coded $H(Q(E_i))$. The decoder uses a singleton architecture to decompress the pyramid of features. The decoder $D_i(\theta_{k:1 \leq k \leq k-1})$ 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 $z$ is computed as $\hat{z} = Q(z) = z + U\left(-\frac{1}{2}, \frac{1}{2}\right)$

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

$$p_{\hat{z}|\psi} = \prod_i \left( p_{z_i|\psi}(\psi^i) \ast U\left(-\frac{1}{2}, \frac{1}{2}\right) \right) (\hat{z}_i) \quad (1)$$

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

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 $D$ can be seen as a group of hierarchical networks with shared weights where each network $D_i$ learns the additional set of trainable parameters $\theta_i$ along with the parameters of $D_{i-1}(\theta_{k:1 \leq k \leq k-1})$ 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 $P$ is computed as the Mean Squared Error (MSE) between the
features of pre-trained VGG-19 architecture, and the distortion \( d(\cdot, \cdot) \) is computed using MSE between ground truth and reconstructed image. The overall loss function is given as,
\[
L(\theta, \phi) = \sum_i \alpha_i \mathbb{E}[d(x, D_i(Q(E_i(x; \theta)); \phi_1 \leq k \leq i))] \\
+ \sum_i \beta_i \mathbb{E}[P(x, D_i(Q(E_i(x; \theta)); \phi_1 \leq k \leq i))] + \sum_i \gamma_i H_i
\]

In equation 2, the index \( i \) represents the \( i^{th} \) level in the feature pyramid, and the parameters \( \alpha_i \), \( \beta_i \) and \( \gamma_i \) are the Lagrangian multipliers to control the trade-off among distortion \( d \), perceptual quality \( P \) 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 \( h_2 = GDN(W_{h1}) \). 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 \( W \neq 0 \), \( h_1 \) 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 \( R = (\frac{n^2}{m+\lambda_0}) \) with probability at least \( 1-\delta \), where \( m,n \) are number of parameters in network and number of training samples respectively, and \( \lambda_0 \) is the smallest eigenvalue of gram matrix \( H_{ij} \) given by
\[
H_{ij} = \mathbb{E}_{u \sim N(0, I)}[(h_i)^T h_j] \mathbb{1}_{(u^T h_i \geq 0, u^T h_j \geq 0)}
\]
To avoid the equality, the weights must stay outside of hyper-sphere of radius \( R \). Solving for \( W \), the learned weights will lie out of this hyper-sphere for
\[
(\frac{n^2}{m+\lambda_0})^{\frac{1}{2}} \geq (GDN^{-1}(h_2)^T h_1^{\frac{1}{2}})^{-1} - w(0)
\]
number of parameters in the network with probability at least \( 1-\delta \). 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 \( 5 \times 10^{-4} \) after MSE loss reaches 330. The training images are cropped to \( 240 \times 240 \) with a batch size of 16.

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

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

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.

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


