Simplified entropy model for reduced-complexity end-to-end variational autoencoder with application to on-board satellite image compression

Vinicius Alves de Oliveira, Thomas Oberlin, Marie Chabert, Charly Poulliat, Bruno Mickael, Christophe Latry, Mikael Carlavan, Simon Henrot, Frederic Falzon, Roberto Camarero

SIMPLIFIED ENTROPY MODEL FOR REDUCED-COMPLEXITY END-TO-END VARIATIONAL AUTOENCODER WITH APPLICATION TO ON-BOARD SATELLITE IMAGE COMPRESSION

V. Alves de Oliveira[1,2], **T. Oberlin**[3], **M. Chabert**[1], **C. Poulliat**[1], **M. Bruno**[4], **C. Latry**[4], **M. Carlavan**[5], **S. Henrot**[5], **F. Falzon**[5], **and R. Camarero**[6]

[1]*University of Toulouse IRIT / INP-ENSEEIHT*
[2]*TéSA Toulouse*
[3]*University of Toulouse ISAE-SUPAERO*
[4]*CNES Toulouse*
[5]*Thales Alenia Space Cannes*
[6]*ESA Noordwijk*

ABSTRACT

In recent years, neural networks have emerged as data-driven tools to solve problems which were previously addressed with model-based methods. In particular, image processing has been largely impacted by convolutional neural networks (CNNs) [B$^+$09]. Recently, CNN-based autoencoders have been successfully employed for lossy image compression [BLS17, TSCH17, RB17, BMS$^+$18]. These end-to-end optimized architectures are able to dramatically outperform traditional compression schemes in terms of rate-distortion trade-off. The autoencoder first applies an analyzing transform to the input data to produce a latent representation with minimum entropy after quantization. The latent representation, derived through several convolutional layers composed of filters and activation functions, is multi-channel (the output of a particular filter is called a channel or a feature) and non-linear. The representation is then quantized to produce a discrete-valued vector. A standard entropy coding method losslessly compress this discrete-valued vector from a model of the representation probability distribution. The analyzing transform and the representation distribution model are both learned from the data by minimization of a rate-distortion trade-off. The assumed representation probability distribution, leading to a particular entropy model, is a key element of these frameworks. In earlier works [BLS17, TSCH17, RB17], the learned representation was assumed independent and identically distributed within each channel and the channels were assumed independent of each other, resulting in a fully-factorized entropy model. Moreover, a fixed entropy model was learned once, from the training set, prohibiting adaptation to the input image during the operational phase. The variational autoencoder proposed in [BMS$^+$18] used an auxiliary autoencoder. This autoencoder estimates the hyper-parameters of the representation distribution, for each input image even in operational phase. It does not require the assumption of a fully-factorized model which conflicts with the need for context modeling. This variational autoencoder achieves compression performance close to the one of BPG (Better Portable Graphics) [Bel15] at the expense of a considerable increase in complexity. However, in the context of on-board compression, a trade-off between compression performance and complexity has to be considered to take into account the strong computational constraints. For this reason, the computational requirements of the CCSDS (Consultative Committee for Space Data Systems) [Boo05] have been considerable reduced with respect to JPEG2000. This work follows the same logic, however in the context of learned image compression. The aim of this paper is to design a simplified version of the variational autoencoder proposed in [BMS$^+$18] in order to meet the on-board constraints in terms of complexity while preserving high performance in terms of rate-distortion. Apart from straightforward simplifications of the transform (e.g. reduction of the number of filters in the convolutional layers), we mainly propose a simplified entropy model that preserves the adaptability to the input image. A preliminary reduction of the number of filters reduces the complexity by 62 % the number of floating point operations per pixel (FLOPp) with respect to [BMS$^+$18]. It also reduces the number of learned parameters with a positive impact on the memory occupancy. The entropy model simplification exploits a statistical analysis of the learned representation for satellite images, also performed in [DRG18] for natural images. This analysis reveals that most of the features are well fitted by centered Laplacian distributions and by Gaussian distributions to a lesser extent. The computationally expensive auxiliary autoencoder in [BMS$^+$18] is replaced by a classical and simple estimation of a single parameter referred to as the scale (resp. the standard deviation). Our simplified entropy model reduces the complexity of the variational autoencoder coding part by 18 % and outperforms the autoencoder proposed in [BLS17] for the relatively high target rates.

Key words: image compression; neural networks; transform coding.

1. INTRODUCTION

The spatial resolution of Earth observation satellite images regularly increases thanks to the sensor technological evolution. In order to save transmission channel bandwidth, memory storage and data-transmission time, on-board compression needs to cope up with this increasing volume of digital data [Hua11]. Compression techniques can be classified into two categories: lossless and lossy compression. Lossless compression is a reversible technique that compress data without loss of information. However, lossless compression rates are limited: for optical satellite images, the typical lossless compression rate that can be achieved is less than 3:1 [Qia13]. Lossy compression achieves high compression rates through transform coding [Goy01] and the optimization of a rate-distortion criterion. Traditional frameworks for lossy image compression operate by linearly transforming the data into an appropriate continuous-valued representation, quantizing its coefficients independently, and then encoding this discrete representation using a lossless entropy coder. To give an on-ground example, JPEG uses a discrete cosine transform (DCT) on blocks of pixels followed by a Huffman coder whereas JPEG2000 uses an orthogonal wavelet decomposition followed by an arithmetic coder. In the context of on-board compression, the consultative committee for space data systems (CCSDS), drawing on the on-ground JPEG2000 standard, recommends the use of the orthogonal wavelet transform. However, the computational requirements of the CCSDS have been considerably reduced with respect to JPEG2000, taking into account the huge hardware constraints on satellites. This work follows the same logic, however in the context of learned image compression. In recent years, neural networks have emerged as powerful data-driven tools to solve problems previously addressed with model-based methods. In particular, image processing has been largely impacted by convolutional neural networks (CNNs). CNNs have proven to be successful in many computer vision applications [B$^+$09] such as classification [HSS18], object detection [RDGF16], segmentation [KWL$^+$17], denoising [ZZC$^+$17] and feature extraction [WB17]. Recently, CNNs have been successfully employed for lossy image compression [BLS17, TSCH17, RB17, BMS$^+$18]. CNN end-to-end optimized architectures are able to dramatically outperform traditional compression schemes in terms of rate-distortion trade-off, however at the cost of a high computational complexity. In this paper, we start from the state-of-the-art CNN image compression scheme proposed in [BMS$^+$18] to design a reduced-complexity framework in order to adapt to satellite image compression. Apart from a straightforward network reduction, we mainly propose a simplified entropy model that preserves the adaptability to the input image. The entropy model simplification exploits a statistical analysis of the learned representation for satellite images, also performed in [DRG18] for natural images. The paper is organized as follows. Section 2 presents some background on learned image compression. Section 3 presents the proposed reduced-complexity compression scheme. Section 4 quantitatively assesses its performance on a set of real satellite images and performs a comparative complexity study. Section 5 concludes the paper.

2. LEARNED COMPRESSION BACKGROUND

2.1. Autoencoder structure

Autoencoders have been initially designed for data dimension reduction similar to e.g. Principal Component Analysis (PCA) [B$^+$09]. In the case of image compression, the goal is not only to reproduce the image, but to produce a representation with low entropy after quantization. The autoencoder is composed of an analyzing transform denoted by G_a and a synthesis transform denoted by G_s both learned from the data. The interface between the analysis transform and the synthesis transform is called the bottleneck. In the case of compression, this bottleneck is composed of a quantizer, an entropy encoder and its associated decoder. The analysis transform is applied to the input data $\mathbf{x}$ to produce a representation $\mathbf{y} = G_a(\mathbf{x})$. This representation, derived through several convolutional layers composed of filters and activation functions, is multi-channel (the output of a particular filter is called a channel or a feature) and non-linear. At the input of the bottleneck, the representation is quantized to produce a discrete-valued vector $\hat{\mathbf{y}} = Q(\mathbf{y})$ with minimum entropy. A standard entropy coding method, such as arithmetic, range or Huffman coding [RL81] uses the entropy model inferred from the representation, denoted by $p_{\hat{\mathbf{y}}}(\hat{\mathbf{y}})$, to losslessly compress $\hat{\mathbf{y}}$. The last step consists in decoding and transforming back the quantized representation into the original space: $\hat{\mathbf{x}} = G_s(\hat{\mathbf{y}})$ by means of the synthesis transform, composed of so-called transpose convolutional layers. In [BLS17, BMS$^+$18] the activation functions in G_a (resp. G_s) are generalized divisive normalizations (GDN) (resp. Inverse Generative Divisive Normalizations (IGDN)). Contrarily to usual activation functions (e.g. ReLU, sigmoid,...), GDN and IGDN are parametric functions that implement an adaptive normalization. The learning of their parameters is thus required. They have been shown to reduce statistical dependencies [Bal18, ML10, Lyu10] and thus appear particularly appropriate for transform coding. According to [Bal18], the GDN better estimates the optimal transform than conventional non-linearities for a wide range of rate-distortion trade-offs. Although the GDN increases the number of parameters to be learned, this increase represents a percentage of less than 2% of the overall structure with respect to conventional non-linearities. For that purpose, our complexity reduction does not target the GDN/IGDN.

2.2. Autoencoder learning

The autoencoder parameters (filter weights, GDN/IGDN and representation distribution model parameters) are jointly learned by optimization of a loss function that establishes a trade-off between the rate $R(\hat{\mathbf{y}})$ and the distortion $D(\mathbf{x}, \hat{\mathbf{x}})$ between the original image $\mathbf{x}$ and the reconstructed image $\hat{\mathbf{x}}$. The rate-distortion criterion then writes as the weighted sum:

$$J = \lambda D(\mathbf{x}, \hat{\mathbf{x}}) + R(\hat{\mathbf{y}}), \tag{1}$$

where parameter λ tunes the rate-distortion trade-off. The rate achieved by an entropy coder is lower-bounded by the entropy derived from the actual discrete probability distribution of the quantized vector $\hat{\mathbf{y}}$, denoted as $m(\hat{\mathbf{y}})$. The rate increase comes from the mismatch between the probability distribution model $p_{\hat{\mathbf{y}}}(\hat{\mathbf{y}})$ required for the coder design and $m(\hat{\mathbf{y}})$: the smallest bit-rate is given by the Shannon cross entropy between the two distributions:

$$H(\hat{\mathbf{y}}) = \mathbb{E}_{\hat{\mathbf{y}} \sim m} \left[-log_2 p_{\hat{\mathbf{y}}}(\hat{\mathbf{y}}) \right]. \tag{2}$$

The probability model $p_{\hat{\mathbf{y}}}(\hat{\mathbf{y}})$ has thus a strong impact on the coder performance.
The distortion measure is chosen to account for image quality as perceived by a human observer. Due to its many desirable computational properties, the mean square error (MSE) is generally selected. However, a measure of perceptual distortion may also be employed such as the multi-scale structural similarity index (MS-SSIM) [WSB03]. The autoencoder parameters are learned by minimizing the loss function defined in Equation 1 through gradient descent with backpropagation [B$^+$09] on a representative image training set. However, this requires the loss function to be differentiable. In the specific context of compression, a major hurdle is that the derivative of the quantization function is zero everywhere except at integers, where it is undefined. To overcome this difficulty, Ballé et al. (2016) [BLS17] proposed to replace the quantization by the addition of an independent and identically distributed (i.i.d.) uniform noise, while Theis et al. (2017) [TSCH17] proposed to replace, in the backward pass (i.e. when back-propagating the error), the derivative of the quantization function with a smooth approximation. In both cases, the quantization is kept as it is in the forward pass (i.e. when processing an input data). Note that, in the following, we will consider the first approach [BLS17].

2.3. Entropy model

As stressed above, a key element in the end-to-end learned image compression frameworks is the entropy model defined through the probability distribution model $p_{\hat{\mathbf{y}}}(\hat{\mathbf{y}})$ assigned to the quantized representation for coding. The bit-rate is minimized if the distribution model $p_{\hat{\mathbf{y}}}(\hat{\mathbf{y}})$ is equal to the actual distribution $m(\hat{\mathbf{y}})$. Unfortunately, this distribution may differ from the actual unknown distribution of the quantized representation $m(\hat{\mathbf{y}})$, arising from the actual distribution of the input image and from the analysis transform G_a.

As mentionned previously, for back-propagation derivation during the training step, Ballé approximates the quantization process ($\hat{\mathbf{y}} = Q(\mathbf{y})$) by the addition of an i.i.d uniform noise $\Delta\mathbf{y}$, whose range is defined by the quantization step. Due to the adaptive local normalization performed by GDN non-linearities, the quantization step can be set to one without loss of generality. Hence the quantized representation $\hat{\mathbf{y}}$, which is a discrete random variable taking values in $\mathbb{Z}$, is modelled by the continuous random vector

$$\tilde{\mathbf{y}} = \mathbf{y} + \Delta\mathbf{y} \tag{3}$$

taking values in $\mathbb{R}$.
In [BLS17, TSCH17], this approximated quantized representation was assumed independent and identically distributed within each channel and the channels were assumed independent of each other, resulting in a fully-factorized distribution:

$$p_{\tilde{\mathbf{y}}|\psi}(\tilde{\mathbf{y}}|\psi) = \prod_i p_{\tilde{y}_i|\psi^{(i)}}(\tilde{y}_i), \tag{4}$$

where index i runs over all elements of the representation, through channels and spatial locations, $\psi^{(i)}$ is the distribution model parameter vector associated to each element. The addition of the uniform quantization noise leads to the following expression for $p_{\tilde{y}_i|\psi^{(i)}}(\tilde{y}_i)$ defined through a convolution by a uniform distribution on the interval $[-1/2, 1/2]$:

$$p_{\tilde{y}_i|\psi^{(i)}}(\tilde{y}_i) = p_{y_i|\psi^{(i)}}(y_i) * \mathcal{U}(-1/2, 1/2). \tag{5}$$

In [BLS17, TSCH17], the parameter vectors are learned from data during the learning phase. This learning, performed once and for all, prohibits adaptivity to the input images during operational phase. Moreover, the simplifying hypothesis of a fully factorized distribution is very strong. In particular, elements of $\hat{\mathbf{y}}$ show spatial dependency in practice, as observed in [BMS$^+$18].
To overcome these limitations and thus to obtain a more realistic and more adaptive entropy model, [BMS$^+$18] proposed a variational autoencoder which takes into account possible spatial dependency in each input image. Auxiliary random variables $\tilde{\mathbf{z}}$, conditioned on which the quantized representation $\tilde{\mathbf{y}}$ elements are independent, are derived from $\mathbf{y}$ by an additional autoencoder, connected in parallel with the bottleneck. The hierarchical model hyper-parameters are learned for each input image in operational phase. Firstly, the hyperprior transform analysis H_a produces the set of auxiliary random variables $\mathbf{z}$. Secondly, $\mathbf{z}$ is transformed by the hyperprior synthesis transform H_s into a second set of random variables σ.
In [BMS$^+$18], $\mathbf{z}$ distribution is assumed fully-factorized and each representation element $\tilde{y}_i$, knowing $\mathbf{z}$, is modeled by a zero-mean Gaussian with its own standard deviation σ_i. Finally, taking into account the quantization process, the conditional distribution of each quantized representation element is given by:

$$\tilde{y}_i|\tilde{\mathbf{z}} \sim \mathcal{N}\left(0, {\sigma_i}^2\right) * \mathcal{U}\left(-\frac{1}{2}, \frac{1}{2}\right). \tag{6}$$

The rate computation involves this distribution model as well as the prior distribution of $\tilde{\mathbf{z}}$, that will have to be transmitted to the decoder with the compressed data, as side information. This variational autoencoder allows to reach state-of-the-art compression performance, close to the one of BPG (Better Portable Graphics) [Bel15] at the expense of a considerable increase in complexity with respect to [BLS17], reflected by a runtime increase between 20% and 50%. The two reference architectures [BLS17] and [BMS$^+$18] are compactly displayed on Figure 1. The left column represents the autoencoder [BLS17] whereas the combination of left and right columns represents the variational autoencoder [BMS$^+$18]. N represents the number of filters which is common to all layers except the ones directly connected to the bottleneck. These layers are generally composed of M filters with $M > N$ to obtain a so-called wide bottleneck [CSTK19], for increased performance.

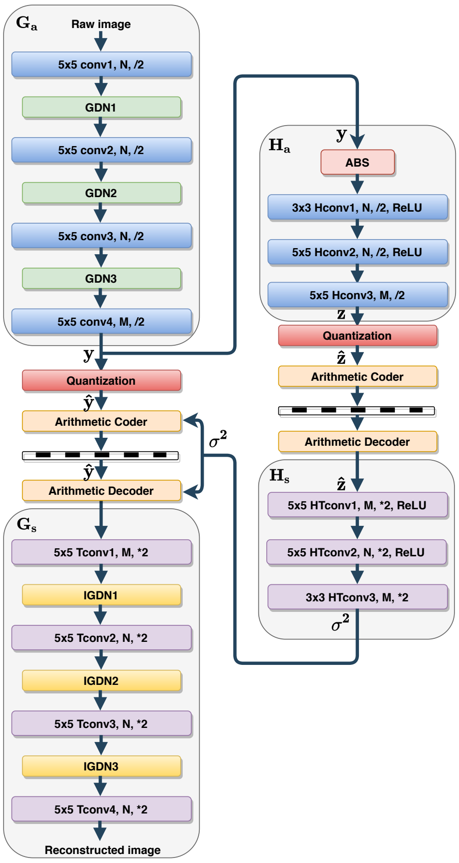


Figure 1. Architecture of the autoencoder [BLS17] (left column) and variational autoencoder [BMS$^+$18] (left and right columns).

3. REDUCED-COMPLEXITY VARIATIONAL AUTOENCODER

In the literature, the design of learned image compression frameworks hardly takes into account the computational complexity: the objective is merely to obtain the best performance in terms of rate/distortion. However, in the context of on-board compression, a trade-off between performance and complexity must be considered to take into account the strong computational constraints. Our focus here is to propose a complexity-reduced alternative to the state-of-the-art structure [BMS$^+$18] while minimizing the impact on the rate-distortion performance. Note that this complexity reduction is particularly relevant when it operates on the coding part of the framework, the one subject to the on-board constraints.

3.1. Reduction of the number of filters

A straightforward simplification to be first considered is the reduction of the number of filters composing the convolutional layers. The state-of-the-art frameworks generally involve a large number of filters with the objective of increasing the network approximation capacity, which is needed at high bit rates [BLS17, BMS$^+$18]. However, the learning of the associated parameters (e.g. the filter coefficients) is computationally demanding, in addition to the increased memory it requires. Moreover, a reduction in the number of filters indirectly implies a reduction in the number of parameters and operations in the GDN/IGDN, as it reduces the depth of the tensors (equal to the number of filters) at their input. However, the impact of reducing the number of filters depends on the convolutional layers it applies. Indeed, a high number of filters M has to be maintained in the last layer of the encoder and in the first layer of the decoder, following the so-called wide bottleneck strategy [CSTK19]. For the other layers, the number N of filters can be subsequently reduced. The same strategy benefits to the auxiliary autoencoder implementing the hyperprior illustrated in Figure 1 (right column). We consider the number of parameters N_p and the floating point operations per pixel (FLOPp) to compare the proposed and reference frameworks in terms of complexity. These quantities are provided in [CSTK19] for the convolution layers including GDN/IGDN. Quantization is parameter free.

3.2. Simplified parametric entropy model

The entropy model simplification aims at achieving a compromise between simplicity and performance while preserving the adaptability to the input image. In [BLS17], the representation distribution is assumed fully-factorized and the statistical model for each feature is non-parametric to avoid the a priori choice of a given distribution shape. Note that ”non-parametric” means that the shape of the distribution is not known in advance, however it is defined by some parameters. These parameters are learned once, during the learning step. In [BMS$^+$18], the strong independence assumption leading to a fully-factorized model is avoided by the introduction of the hyperprior distribution, whose parameters are learned for each input image even in the operational phase. Both models are general and thus suitable to a wide variety of images, however the first one implies a strong hypothesis of independence and prohibits adaptivity while the second one is computationally expensive. This paper exploits a statistical analysis of each feature of the learned representation in the particular case of real satellite images. A similar statistical analysis has been previously conducted in the case of natural images in [DRG18] with the objective to properly design the quantization step in [BLS17]. The probability density function related to each feature was estimated through a normalized histogram from a representative set of natural images. The study showed that most features can be accurately modelled as Laplacian random variables. A similar result has also been obtained in [LG00] for block-DCT coefficients of natural images under the assumption that the variance is constant on each image block and that its values on the different blocks are distributed according to an exponential or an halfnormal distribution. We conducted this statistical analysis on the representation obtained by the end-to-end framework [BLS17] when trained on a representative dataset of satellite images and for the highest rates obtained with this framework (between 2.5bpp and 3bpp as displayed on Figure 5). According to Kolmogorov-Smirnov goodness-of-fit test [PG81], most features follow a Laplacian distribution defined by:

$$f(\zeta, \mu, b) = \frac{1}{2\lambda}\left(-\frac{|\zeta - \mu|}{b}\right) \text{ for } \zeta \in \mathbb{R}. \qquad (7)$$

where μ is the mean value and $b > 0$ is a scale parameter related to the variance by $Var(\zeta) = 2b^2$. As an illustration, let consider the satellite image displayed on Figure 2. This image of the city of Cannes (French Riviera) is a 12-bit simulated panchromatic Pléiade image with size 512×512 and resolution 70cm. According to the Kolmogorov-Smirnov goodness-of-fit test [PG81], 93% of the features follow a Laplacian distribution with a significance level $\alpha = 5\%$. Figure 3 shows a particular 32×32 feature derived from the Cannes image and its normalized histogram with Laplacian fitting.

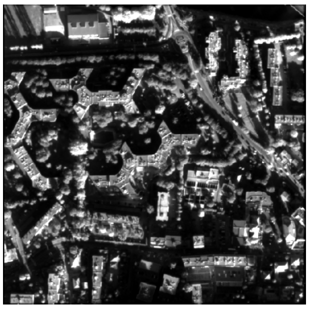
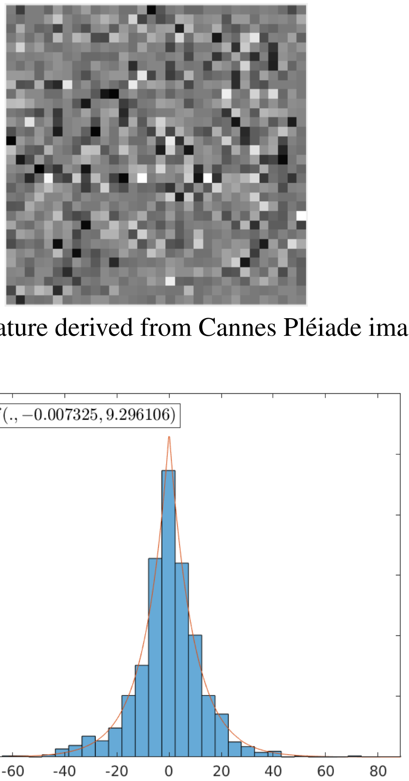


Figure 2. Simulated 12-bit Pléiade image of Cannes with size 512×512 *and resolution* $70cm$


(a). One feature derived from Cannes Pléiade image

(b). Associated normalized histogram and Laplacian fitting

Figure 3. An example of feature from Cannes Pléiade image and its normalized histogram with Laplacian fitting.

This statistical analysis has been performed for each of the 16 simulated 512×512 Pléiade images of the test set (the set used for performance analysis in section 4). For all images, most of the features can be modelled by Laplacian random variables with relatively small mean value and scale varying according to the considered feature and to the kind of image. Note that the Gaussian fitting is also appropriate albeit to a lesser extent. We thus propose to consider the following parametric model for all the j^{th} feature elements, denoted as y_{i_j} for $i_j \in I_j$, where I_j denotes the set of indexes covering the j^{th} feature:

$$\begin{aligned} y_{i_j} &\sim \text{Laplace}(0,\, b_j) \text{ (resp. } y_i^j \sim \mathcal{N}(0,\, \sigma_j^2)) \\ \text{with: } b_j &= \sqrt{Var(y_i^j)/2} \text{ (resp. } \sigma_j^2 = Var(y_i^j)). \end{aligned} \qquad (8)$$

The problem then boils down to the estimation of a single parameter per feature referred to as the scale b_j (respectively the standard deviation σ_j) in the case of the Laplacian (resp. Gaussian) distribution. Starting from [BMS+18], this proposal reduces the complexity at two levels. First, the hyperprior autoencoder, including the analysis H_a and synthesis H_s transforms, is removed. Second, the side information initially composed of the compressed auxiliary random variable set ($\mathbf{z}$) of size $8 \times 8 \times M$ now reduces to a $M \times 1$ vector of variances. The auxiliary network simplification is displayed on the right part of Figure 4.

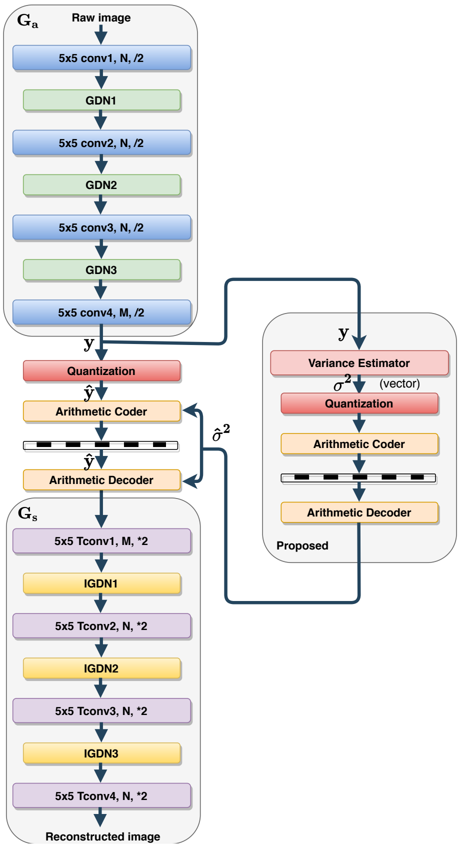


Figure 4. Architecture of the proposed simplified entropy model: autoencoder handling the input image (left column) and proposed simplification of the autoencoder implementing the hyperprior (right column).

4. PERFORMANCE ANALYSIS

4.1. Implementation setup

To assess the relevance of the proposed complexity reductions, experiments were conducted using TensorFlow. The batch size (i.e. the number of training samples to work through before the parameters are updated) was set to 8 and up to 1M iterations were performed. Both training and validation datasets are composed of simulated 12-bit Pléiades panchromatic images provided by the CNES, covering various landscapes (i.e. desert, water, forest, industrial, cloud, port, rural, urban). The training dataset is composed of 8M of patches (of size 256×256) randomly cropped from 112 images (of size 585×585). The validation dataset is composed of 16 images (of size 512×512). MSE was considered as the distortion metric for training. The rate and distortion measurements were averaged across the validation dataset for a given value of λ. In addition to the MSE, we also evaluate those results in terms of MS-SSIM. Note that they exhibit a similar behavior even if the models were trained for the MSE only. The proposed framework is compared with the CCSDS 122.0-B [Boo05], JPEG2000 and with the reference methods AE-N128 [BLS17] and VAE-N192 [BMS$^+$18].

4.2. Impact of the number of filter reduction

Starting from [BMS$^+$18] denoted as VAE-N192, the number of filters is reduced from $N = 192$ to $N = 64$ for all layers, in the main autoencoder and in the hyperprior one, except from the two layers on both sides of the bottleneck composed of $M = 192$ filters. The resulting architecture is termed VAE-N64. The complexity of this model is detailed in Table 1.

Table 1. Detailed complexity of VAE-N64

Layer	**Kernel**		**Channel**		**Output**		**Np**	**FLOPp**
	h	w	C_{in}	C_{out}	$H^{'}$	$W^{'}$		
conv1	5	5	1	64	128	128	1664	4.12×10^2
GDN1							4160	1.03×10^3
conv2	5	5	64	64	64	64	102464	6.39×10^3
GDN2							4160	2.59×10^2
conv3	5	5	64	64	32	32	102464	1.58×10^3
GDN3							4160	2.59×10^2
conv4	5	5	64	192	16	16	307392	1.19×10^3
Hconv1	3	3	192	64	16	16	110656	4.27×10^2
Hconv2	5	5	64	64	8	8	102464	0.91×10^2
Hconv3	5	5	64	192	4	4	307200	0.61×10^2
HTconv1	5	5	192	192	8	8	921792	9.00×10^2
HTconv2	5	5	192	64	16	16	307264	1.19×10^3
HTconv3	3	3	64	192	16	16	110784	4.27×10^2
Tconv1	5	5	192	192	32	32	921792	1.43×10^4
IGDN1							37056	5.79×10^2
Tconv2	5	5	192	64	64	64	307264	1.92×10^4
IGDN2							4160	2.59×10^2
Tconv3	5	5	64	64	128	128	102464	2.55×10^4
IGDN3							4160	1.03×10^3
Tconv4	5	5	64	1	256	256	1601	1.60×10^3
Total							3765118	7.69×10^4

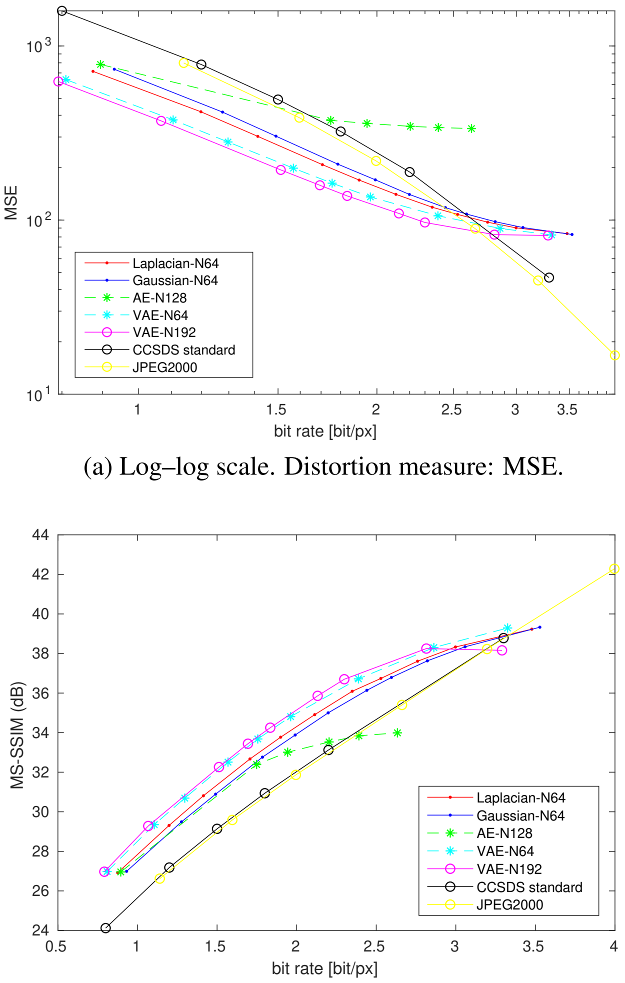
Compared with VAE-N192, the VAE-N64 complexity is 62% lower in terms of FLOPp, as displayed in Table 2. Note that the subsequent reduction of the number of learned parameters has also a positive impact on the memory occupancy. Moreover, as displayed on Figure 5, VAE-N64 achieves a rate-distortion performance close to the one of VAE-N192 [BMS$^+$18], both in terms of MSE and MS-SSIM, even at relatively high rates, contrary to what was stated in [BLS17, BMS$^+$18, Bal18].

4.3. Impact of the entropy model simplification

The proposed simplified models Laplacian-N64 and Gaussian-N64 are compared with the original VAE-N192 of [BMS$^+$18], with VAE-N64, and with the non-variational method AE-N128 [BLS17]. Figure 5 shows

Table 2. Comparative complexity of the global architectures

Method	**Np**	**FLOPp**	**Relative**
VAE-N192	9889793	1.98×10^5	1.00
VAE-N64	3765118	7.69×10^4	0.38
Laplacian/Gaussian-N64	1904958	7.37×10^4	0.371



(a) Log–log scale. Distortion measure: MSE.

(b) Distortion measure: MS-SSIM (dB).

Figure 5. Rate-distortion curves for the considered learned frameworks and for the CCSDS and JPEG2000 standards for MSE and MS-SSIM (dB) (derived as $-10\log_{10}(1-\text{MS-SSIM})$*).*

the rate-distortion averaged over the validation dataset for the trained models for both MSE and MS-SSIM quality measures. Recall that the architecture was trained for MSE only. The proposed simplified entropy model achieves an intermediate performance between the variational model with a reduced number of filters (VAE-N64) and the non-variational model AE-N128 [BLS17]. Obviously, due to the entropy model simplification, Laplacian-N64 and Gaussian-N64 underperform the more general and thus more complex VAE-N64 model. However, the proposed entropy model, even if simpler, preserves the adaptability to the input image, unlike the model AE-N128 [BLS17]. Note that the simplified entropy models perform close to the hyperprior models at relatively high rates such as targeted for satellite image compression. One possible explanation for this behaviour can be the increased amount of side information required by the hyperprior model [BMS$^+$18] for increasing rates [HYML20]. Table 3 shows that the coding part complexity of Laplacian-N64 and Gaussian-N64 is 22% lower than the one of Hyperprior-N64.

Table 3. Reduction of the encoder complexity induced by simplified entropy model on the coding part

Method	Np	FLOPp	Relative
VAE-N64	2386621	1.42×10^4	1
Laplacian/Gaussian-N64	526461	1.11×10^4	0.78

5. CONCLUSION

By combining the reduction of the number of filters, and by proposing a simplified entropy model, we have developed a reduced-complexity compression architecture for satellite images that outperforms the CCSDS 122.0-B [Boo05] while maintaining a competitive performance for medium to relatively high rates in comparison with the previous learned image compression models [BLS17, BMS$^+$18]. In future research, we plan to improve the performance of the learned satellite compression models at high bit-rates and high resolutions by fine-tuning the size of the network (e.g. number of filters) according to the target rates. Preliminary results are given in Figure 5, for MSE distortion measure only, the MS-SSIM shows the same behaviour. They have been obtained by an increase of the number of filters on both sides of the bottleneck only (from $M = 192$ to $M = 256$) keeping $N = 64$. Theses curves show that the simplified learned architecture outperforms the JPEG2000 and CCSDS standard even at higher rates subject to a fine-tuning of the network dimensioning, in each rate range, for a good trade-off between performance and complexity.

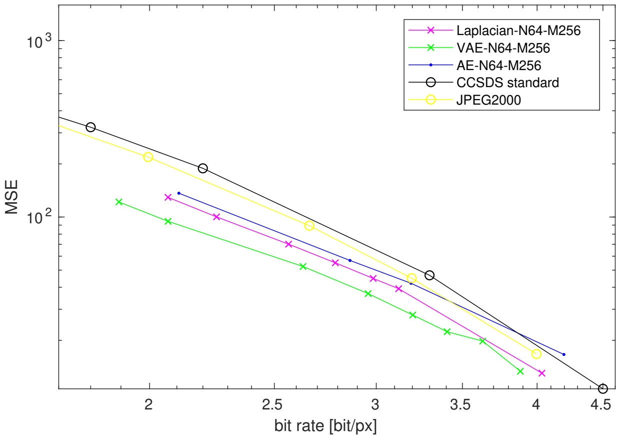


Figure 6. Rate-distortion curves at higher rates for learned frameworks and for the CCSDS and JPEG2000 standards for MSE in log-log scale.

ACKNOWLEDGMENTS

This work has been carried out under the financial support of the French space agency CNES and Thales Alenia Space.

REFERENCES

[B⁺09] Yoshua Bengio et al. Learning deep architectures for ai. *Foundations and trends in Machine Learning*, 2(1):1–127, 2009.

[Bal18] Johannes Ballé. Efficient nonlinear transforms for lossy image compression. In *2018 Picture Coding Symposium (PCS)*, pages 248–252. IEEE, 2018.

[Bel15] Fabrice Bellard. Bpg image format. *URL https://bellard. org/bpg*, 1, 2015.

[BLS17] Johannes Ballé, Valero Laparra, and Eero Simoncelli. End-to-end optimized image compression. *International Conference on Learning Representations*, 2017.

[BMS⁺18] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *International Conference on Learning Representations*, 2018.

[Boo05] Blue Book. *Consultative Committee for Space Data Systems (CCSDS), Image Data Compression CCSDS 122.0-B-1, ser. Blue Book, Nov. 2005.* CCSDS Secretariat, 2005.

[CSTK19] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Deep residual learning for image compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[DRG18] Thierry Dumas, Aline Roumy, and Christine Guillemot. Autoencoder based image compression: can the learning be quantization independent? In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1188–1192. IEEE, 2018.

[Goy01] Vivek K Goyal. Theoretical foundations of transform coding. *IEEE Signal Processing Magazine*, 18(5):9–21, 2001.

[HSS18] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[Hua11] Bormin Huang. *Satellite Data Compression*. Springer Science & Business Media, 2011.

[HYML20] Yueyu Hu, Wenhan Yang, Zhan Ma, and Jiaying Liu. Learning end-to-end lossy image compression: A benchmark. *arXiv preprint arXiv:2002.03711*, 2020.

[KWL⁺17] Pascal Kaiser, Jan Dirk Wegner, Aurélien Lucchi, Martin Jaggi, Thomas Hofmann, and Konrad Schindler. Learning aerial image segmentation from online maps. *IEEE Transactions on Geoscience and Remote Sensing*, 55(11):6054–6068, 2017.

[LG00] Edmund Y Lam and Joseph W Goodman. A mathematical analysis of the dct coefficient distributions for images. *IEEE transactions on image processing*, 9(10):1661–1666, 2000.

[Lyu10] Siwei Lyu. Divisive normalization: Justification and effectiveness as efficient coding transform. In *Advances in neural information processing systems*, pages 1522–1530, 2010.

[ML10] Jesús Malo and Valero Laparra. Psychophysically tuned divisive normalization approximately factorizes the pdf of natural images. *Neural computation*, 22(12):3179–3206, 2010.

[PG81] John W Pratt and Jean D Gibbons. Kolmogorov-smirnov two-sample tests. In *Concepts of nonparametric theory*, pages 318–344. Springer, 1981.

[Qia13] Shen-En Qian. Optical satellite data compression and implementation. In *SPIE—Int. Soc. Opt. Eng.* SPIE, 2013.

[RB17] Oren Rippel and Lubomir Bourdev. Real-time adaptive image compression. In *International Conference on Machine Learning*, pages 2922–2930, 2017.

[RDGF16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[RL81] Jorma Rissanen and G Langdon. Universal modeling and coding. *IEEE Transactions on Information Theory*, 27(1):12–23, 1981.

[TSCH17] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. *International Conference on Learning Representations*, 2017.

[WB17] Thomas Wiatowski and Helmut Bölcskei. A mathematical theory of deep convolutional neural networks for feature extraction. *IEEE Transactions on Information Theory*, 64(3):1845–1866, 2017.

[WSB03] 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.

[ZZC⁺17] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.