Reduced-Complexity Multirate Remote Sensing Data Compression With Neural Networks

Sebastià Mijares i Verdú, Marie Chabert, *Member, IEEE*, Thomas Oberlin, *Member, IEEE*, and Joan Serra-Sagristà, *Senior Member, IEEE*

***Abstract*— One of the main limitations to the adoption of deep learning for image compression is the need to train multiple models to compress at multiple rates. In the case of onboard remote sensing data compression, another limitation is the computational cost of the neural networks. Addressing both limitations, this letter presents a new reduced-complexity architecture for multirate compression of remote sensing images. The proposed architecture enables compressing at a precise user-selected rate while keeping a competitive performance in lossy compression on different sets of remote sensing data. The proposed approach is amenable for onboard deployment.**

***Index Terms*— Data compression, deep learning, lossy compression, multirate, remote sensing.**

I. Introduction

THERE is a lot to see from above, as illustrated by the approximately 5500 satellites that are currently in orbit around the Earth, of which more than 1100 are dedicated to Earth observation, according to the Union of Concerned Scientists as of May 2022 [1]. With 19% of those Earth observation satellites launched in the last two years, it is clear that interest in remote sensing remains strong today, with ever more data being sensed and requiring transmission down to Earth.

Remote sensing data compression is crucial, given the vast volumes of captured data and the satellite's limited downlink capacity. In particular, lossy compression is often considered in order to fit the bitrate requirements of the mission [2], [3]. Furthermore, computational capabilities are severely limited on board, introducing yet another key requirement to a remote sensing data compression algorithm. As a result, remote sensing data compression is an active field of research, with many new proposals and developments every year.

Manuscript received 24 May 2023; revised 20 August 2023; accepted 10 October 2023. Date of publication 20 October 2023; date of current version 2 November 2023. This work was supported in part by the Institute for Artificial and Natural Intelligence Toulouse (ANITI) under Grant ANR-19-PI3A-0004, in part by the Catalan Government under Grant SGR2021-00643, in part by the Spanish Ministry of Economy and Competitiveness, and in part by the European Regional Development Fund under Grant PRE2019-088824 and Grant PID2021-125258OB-I00 (MINECO/FEDER, UE). *(Corresponding author: Sebastià Mijares i Verdú.)*

Sebastià Mijares i Verdú and Joan Serra-Sagristà are with the Department of Information and Communications Engineering, Universitat Autònoma de Barcelona, 08193 Cerdanyola del Vallès, Spain (e-mail: sebastia.mijares@uab.cat).

Marie Chabert is with IRIT/INP-ENSEEIHT, Université de Toulouse, 31071 Toulouse, France.

Thomas Oberlin is with the ISAE-SUPAERO, Université de Toulouse, 31055 Toulouse, France.

Digital Object Identifier 10.1109/LGRS.2023.3325477

Machine learning (ML) has produced a breakthrough in lossy compression for natural images in the last 6 years, surpassing techniques such as JPEG [4], JPEG 2000 [5], and intraframe HEVC [6] in lossy compression [7], [8], [9], [10], [11], [12]. ML compression has also been applied to remote sensing data [13], [14], [15], [16], [17], [18], [19]. These contributions have employed models presented in [7] and [9] as baseline architectures. Regarding architectures dedicated to single-band images, Alves de Oliveira et al. [15] that applying the architecture proposed in [9] outperforms JPEG 2000 [5] for satellite image lossy compression, and further proposed a reduced-complexity version of that architecture competitive with the baseline models. This reduced-complexity design was later used for compression and denoising of panchromatic satellite images [20] and as part of a 1-D + 2-D framework for on board compression of hyperspectral satellite images [21]. Other works published on ML compression of single-band remote sensing images include those by Xu et al. [22] and by Di et al. [23], both of which are also based on [9].

A crucial barrier for the practical adoption of models like those cited above is that they are trained for a specific rate-distortion trade-off, regulated by a parameter in their loss function. As a result, multiple models have to be trained to allow for compression at multiple rates. This is not only costly to train, but also has computational implications (storing multiple models in memory and loading them on and off for compression at different rates), not to mention the fact that they do not allow for a continuous choice of rates. Numerous authors have tackled this problem in order to propose multirate neural image compression [12], [24], [25], [26], [27], [28].

In this letter, we propose a novel multirate variant of the reduced-complexity compression architecture for remote sensing data from [15]. The proposed method features compression at a user-defined bitrate, a novel capability with respect to other multirate compression neural architectures. To the best of our knowledge, this is the first application of such methods to remote sensing data, as well as the first attempt at complexity reduction of such multirate architectures, and a first in practical compression at a user-defined bitrate. It is demonstrated that the proposed multirate compression architecture performs on par with other more complex existing multirate compression architectures and with the multimodel baseline. In the following, multimodel baseline refers to a model trained multiple times, one for each rate-distortion trade-off.

The rest of this letter is structured as follows. Section II introduces the end-to-end optimized transform coding

paradigm this work is based upon, going into detail on some of the multirate compression techniques proposed to date. Section III describes the proposed ML method and the associated architecture. Section IV reports experimental results. Finally, Section V provides a discussion of our findings.

II. End-to-End Optimized Transform Coding

End-to-end optimized transform coding is the state-of-the-art approach for lossy image compression based on ML. Just like in classical transform coding, it consists in encoding the image by transforming it to a latent domain, quantizing it, and entropy-encoding it. For decompression, the bitstream is entropy-decoded and transformed back to the original image domain. Setting this paradigm apart from classical transform coding, here, two neural networks act as the encoder and decoder transforms, respectively, and are jointly trained to minimize the rate-distortion trade-off [11].

Entropy coding in ML image compression can be achieved, for instance, by an arithmetic coder with some probability distribution known to both coder and decoder. In [7] this distribution is fixed, but rate-distortion performance can be greatly improved by adapting the distribution to the input data. This was investigated in [9], introducing a *hyperprior*, which consists of an additional neural network that processes the latent representation to extract and encode some of its parameters, such as its standard deviation. These parameters have to be encoded and sent to the receiver as side information. We will refer hereafter to the autoencoder architecture presented in [9] as the *Ballé2018 architecture*. This lossy compression paradigm using autoencoders has been refined over time and state-of-the-art ML image codecs today use increasingly complex versions of this concept, such as a Gaussian Mixture entropy model and an expanded residual CNN as the main transform [29], or an Asymmetric Gaussian entropy model with a large hyperprior network [12].

The loss function to be optimized in training by these autoencoders is

$$L(x, \varphi, \psi) = R(\varphi(x)) + \lambda D(x, \varphi \circ \psi(x))$$

where $R(\cdot)$ stands for the rate, $D(\cdot, \cdot)$ stands for the distortion between the original and the reconstructed image, and λ is a parameter set during training that regulates the rate-distortion trade-off. Since the model is optimized for a specific rate-distortion trade-off, different models have to be trained in order to allow for compression at different rates.

To overcome the practical limitation imposed by having to train multiple models to compress images at different rates, a multirate architecture would allow for continuous bitrate choices and require the training of a single model, greatly reducing cost in time and resources.

Modulation is one of the most relevant techniques to achieve learned multirate image compression, and aims to mimic the adjustment of quantization step size performed by classical transform codecs. A *modulated autoencoder* is an autoencoder together with an auxiliary neural network—the modulating neural network—which, given some parameter (in this case λ), adjusts the activations in the main autoencoder network to produce a different output. For autoencoders used in image compression, a simple version of this consists in learning a scaling of the latent representation, which amounts to varying the quantization step of the encoder in relation to the rate-distortion multiplier λ, regulating that tradeoff as in the fixed-rate case. In this method, which we will refer to as *bottleneck modulation*, it is preferable to jointly optimize the main autoencoder and the modulating network, in which case the modulated network practically matched the rate-distortion performance of the multimodel baseline [24], [28].

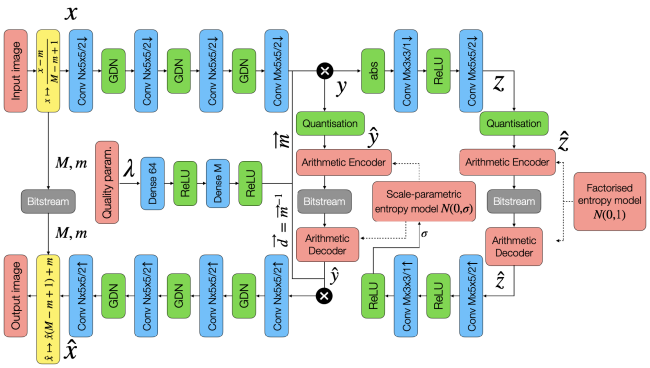

Fig. 1. Proposed architecture. Blocks labeled "Conv $N \times k \times k/s$" indicate convolution with N filters using $k \times k$ kernels with a stride of length s, and the arrow indicates downsampling or upsampling. Blocks labeled "Dense k" indicate a dense feed-forward layer with k nodes. GDN stands for General Divisive Normalization, and ReLU stands for Rectified Linear Unit. The product between a tensor and a vector, $\bigotimes$, is carried out scaling every channel in the tensor by its corresponding entry in the vector.

Various more complex modulated autoencoders have also been proposed, which generally modulate the outputs of every layer in the encoder and decoder, not just the latent representation. We will refer to this approach as *full modulation*. It has been shown that full modulation could more closely match the performance of the multimodel baseline than bottleneck modulation [26]. Full modulation can be extended further to include the parametrization of the entropy model [25].

III. Proposed Method

To address the specific needs of remote sensing data compression using this ML paradigm, we propose a novel method that can be feasibly run on on-board hardware, allows to compress at a user-selected bitrate, and is competitive with current standards and techniques. To the best of our knowledge, this is the first reduced-complexity multirate neural compression method for remote sensing data.

A. Reduced-Complexity Bottleneck-Modulated Compressive Autoencoder

A novel neural network architecture for remote sensing data compression is proposed, based on [9] and shown in Fig. 1. This design's complexity is reduced as in [15] by using a reduced number of filters in the hidden layers. A bottleneck-modulating network [24], [28] is included, which allows us to finely scale the features in the latent space for multirate compression. The proposed architecture also incorporates range-adaptive normalization as proposed in [21]

TABLE I
DETAILED COMPLEXITY OF THE REDUCED-COMPLEXITY BOTTLENECK-MODULATED ARCHITECTURE (PROPOSED AND BALLÉ 2018)

Layer	**Params.** (Ours)	**FLOP/p** (Ours)	**Params.** (Ballé)	**FLOP/p** (Ballé)
Norm.	0	1	0	1
Conv	1,664	416	3,200	800
GDN	4,160	1040	16,512	4,128
Conv	102,464	6,404	409,600	25,600
GDN	4,160	260	16,512	1,032
Conv	102,464	1,601	409,600	6,400
GDN	4,160	65	16,512	258
Conv	307,392	1,201	614,400	2,400
HConv	331,968	1,297	221,184	864
HConv	921,792	900	409,600	400
HConv	0	0	614,400	150
THConv	0	0	614,400	150
THConv	921,792	900	409,600	400
THConv	331,968	1,297	221,184	216
TConv	307,264	1,200	614,400	2,400
iGDN	4,160	65	16,512	65
TConv	102,464	1,601	409,600	6,400
iGDN	4,160	260	16,512	258
TConv	102,464	6,404	409,600	25,600
iGDN	4,160	1,040	16,512	1,032
TConv	1,601	400	3,200	800
Denorm.	0	1	0	1
Dense	128	0	0	0
Dense	12,480	1	0	0
Bott. scaling	0	1	0	0
Total	**3,560,257**	**26,352**	**5,463,040**	**79,355**
Encoder	**3,033,984**	**15,382**	**3,976,704**	**42,799**

for data sources with widely varying sample distributions, as is common for remote sensing images.

The choice of bottleneck-modulation over full modulation as advocated by Yang et al. [26] is motivated by two reasons: complexity reduction and precise bitrate allocation. Regarding complexity, bottleneck modulation clearly requires far fewer operations than full modulation and, although this negatively impacts performance, the difference is small, as is indeed found by Dumas et al. [24] and Yang et al. [26].

B. Complexity Analysis

Just as in [15], we compute the complexity of our proposed architecture and that of the Ballé2018 baseline counting the number of operations per pixel of each layer in Table I. In particular, we specify the number of operations of the encoder since this is the part subject to onboard constraints. Note that the number of filters per layer is detailed in Section IV. Since the modulating network's size is fixed regardless of the input data size, it is assumed in the calculation that the size is 128×128 pixels for the floating-point operations per pixel (FLOP/pixel) calculation. Clearly, the contribution of bottleneck modulation to the number of operations is practically negligible, and the overall complexity and number of parameters in the network is almost identical to that of the equivalent fixed-rate model. As shown in Table I, our proposed method requires 64% fewer operations than the Ballé2018 architecture in encoding.

The number of encoder operations of the proposed architecture of around 15kFLOP/pixel is compatible with an embedded implementation on board using hardware such as the Movidius Myriad 2 [15], and, by extension, with more capable and efficient state-of-the-art hardware, where the method can be run in a short-enough time so that it does not create a backlog of data to be processed. For completeness sake, we also mention that this complexity is two orders of magnitude higher than that of the CCSDS-122.0 or JPEG 2000 standards [30].

C. Precise Bitrate Allocation

Using a bottleneck-modulated network makes precise bitrate allocation computationally practical, unlike using a fully modulated network. This feature is a first in multirate neural image compression. Indeed, the λ parameter used to regulate the rate-distortion tradeoff does not guarantee a fixed bitrate independent of image content. With the Ballé2018 architecture—trained with a single λ value—one may find widely different bitrates for different images depending on their contents. Using a bottleneck-modulated autoencoder, however, the choice of λ may be adjusted iteratively to achieve a user-selected bitrate with a single feed-forward pass. The small computational cost of the modulating network and the scaling of latent representation compared to the rest of the encoding scheme, as illustrated in Table I, makes this viable. This strategy of actively adapting λ to approximate a given bitrate would require multiple feed-forward passes if we used a fully modulated autoencoder, which makes it not feasible in practice.

In the proposed method, precise bitrate allocation is implemented as an iterative process using binary search: starting from the minimum and maximum λ values used in training, $\lambda_{\min}$ and $\lambda_{\max}$, the bitrate at both ends and at their arithmetic mean, $\lambda_{\text{mid}} = ((\lambda_{\min} + \lambda_{\max})/2)$, is computed. If the target bitrate is above that produced by λ_{mid}, $\lambda_{\min} \leftarrow \lambda_{\text{mid}}$ is set, and otherwise $\lambda_{\max} \leftarrow \lambda_{\text{mid}}$ is set, until the bitrate obtained by $\lambda_{\min}$ or $\lambda_{\max}$ is off from the target by some precision error.

IV. EXPERIMENTAL RESULTS

To assess the proposed method, a number of models are trained using the bottleneck-modulated architecture from Fig. 1 or the equivalent fixed-rate multimodel baseline (same backbone transform without modulation). Either mean squared error (mse) or structural similarity index measure (SSIM) are used as distortion metrics in the loss function of the models, optimized using Adam [31]. The reduced-complexity models use $N = 64$ and $M = 192$ filters per layer, while the Ballé2018 architecture is as in [9] ($N = 128, M = 192$), including range-adaptive normalization instead of uniform normalization. The proposed model is compared to said multimodel baseline and to JPEG 2000 [5], and code and visual examples are available at a GitHub repository at the time of submission.[1]

Three different remote sensing datasets are used in our experiments to show the general validity of the proposal as follows.

1) 12-bit simulated panchromatic Pléiades images of 50 cm resolution. A total of $96\,820 \times 820$ images are used in training and $32\,820 \times 820$ images are used in testing.

[1] https://github.com/smijares/mcos2023

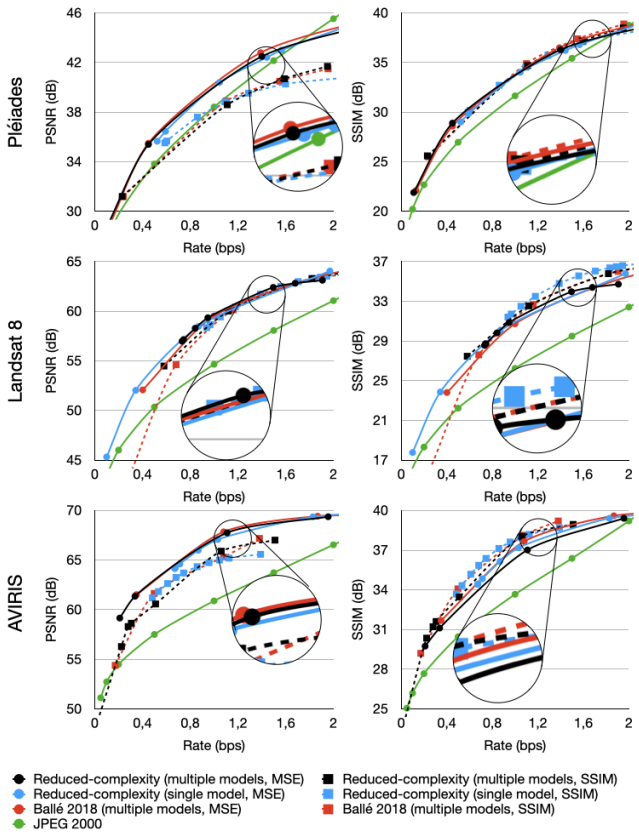

Fig. 2. Rate-distortion performance of our mse and SSIM-optimized models in simulated Pléiades panchromatic images, frame-by-frame Landsat 8 OLI images, and frame-by-frame AVIRIS images. Rate is measured in bits per sample (bps), also known as bits per pixel per component (bpppc).

2) 16-bit L1-processed Landsat 8 OLI images with 30 m spatial resolution, taking frame by frame seven nonpanchromatic bands [32]. A total of 3 584 512 × 512 images are used in training and 1 280 512 × 512 images are used in testing.
3) 16-bit AVIRIS calibrated scenes, taking frame by frame all the 224 spectral bands [33]. The images are of 30 m resolution. A total of 180 scenes are used for training and 20 scenes for testing, in 512 × 512 crops.

These datasets come from panchromatic, multispectral, and hyperspectral images, and have different spatial resolutions, thus, the robustness of the proposed method in spatial decorrelation can be assessed in a variety of conditions. The models are evaluated under the mse and SSIM metrics. Results for mse are converted to peak signal-to-noise ratio (PSNR) as $\text{PSNR} = 20\log_{10}((2^b-1)/(\text{mse})^{1/2})$ dB, with $b = 12$ for Pléiades and $b = 16$ for Landsat 8 and AVIRIS. SSIM results are converted to a decibel scale as $\text{SSIM(dB)} = -10\log_{10}(1-\text{SSIM})$.

Fig. 2 reports the rate-distortion performance of the different models we tested, trained for either mse or SSIM. As is clear from those diagrams, our reduced-complexity bottleneck-modulated models performed on par with the equivalent reduced-complexity multimodel baseline, on par with the Ballé2018 multimodel baseline, and decisively surpassed JPEG 2000 in all datasets, both under PSNR and SSIM as target or evaluation distortion metrics. As expected, our models performed more competitively under the metric they were optimized for. These results show that the architecture simplification from [15] and bottleneck modulation as we propose do not significantly compromise compression performance on a variety of datasets with different characteristics when optimized for the mse and SSIM metrics.

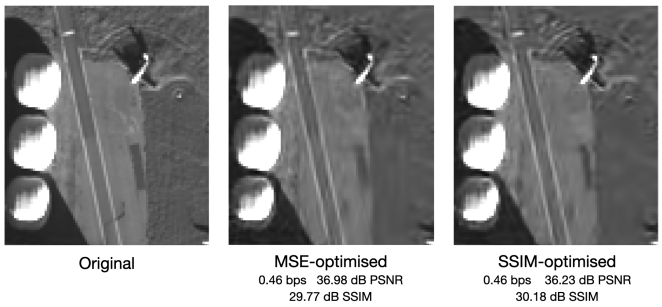

Fig. 3. Visual comparison of a Pléiades image compressed with the mse and SSIM-optimized proposed models at identical bitrate (© CNES 2023).

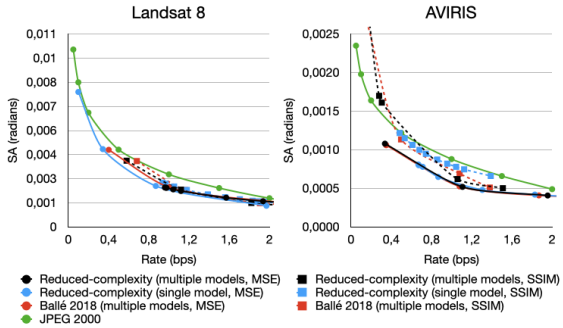

Fig. 4. SA performance of our models in frame-by-frame AVIRIS images and frame-by-frame Landsat 8 OLI images.

Beyond mse and SSIM quantitative metrics, Figs. 3 and 4 provide some qualitative assessment. Fig. 3 shows a visual comparison of a Pléiades image compressed at low bitrate using mse or SSIM-optimized models of the proposed architecture. It can be observed that the SSIM-optimized model recovered certain features more accurately, such as the grooves in the earth near the plane, and generally producing more clearly defined edges. See our Github for more visual examples for Landsat 8 and AVIRIS datasets.

Fig. 4 reports the average spectral angle (SA) loss between the original spectral pixel ($x \in \mathbb{R}^n$) and the distorted spectral pixel ($\hat{x} \in \mathbb{R}^n$), computed as $\text{SA} = \arccos(x^T\hat{x}/\|x\|\|\hat{x}\|)$ for multispectral Landsat 8 and hyperspectral AVIRIS images. As shown, our reduced-complexity single model once more surpasses JPEG 2000 and performs on par with the other learned models. Thus, despite our models only compressing in the 2-D domain, they remain competitive under spectral loss metrics, notably those optimized for mse.

Finally, a runtime comparison between the learned models is carried out to assess the complexity difference theoretically estimated before. The models for Pléiades images are evaluated, measuring the average time to compress all 688 test images. This experiment was conducted on a NVIDIA GeForce RTX 3060 Ti GPU, and Table II lists those times. As expected, the reduced-complexity multimodel baseline was the fastest method, followed by the reduced-complexity modulated model with a single quality input (hence modulating only

TABLE II
LEARNED MODELS RUNTIME COMPARISON

Model	GPU runtime
Reduced-complexity (fixed bitrate)	6.25 s
Ballé2018 (fixed bitrate)	8.90 s
Modulated with single quality input	7.34 s
Modulated with target bitrate	10.09 s

once). Using a target bitrate in the modulated model yielded a slower runtime than the Ballé2018 multimodel baseline. Our multirate proposal requires an average of only 12 iterations to converge to the target bitrate with a precision of ± 0.005 bps.

V. CONCLUSION

A reduced-complexity neural multirate compression architecture for remote sensing data is proposed, which can compress images at multiple and varying bitrates in a single execution, introducing, for the first time, a novel scheme that allows compression at a user-defined bitrate. Experimental results show the proposed method performs on par with the multimodel baseline and is superior to current JPEG 2000 standard in compression of remote sensing images of varying sources and resolutions. Finally, as was the case for the reduced-complexity baseline [15], the proposed encoder could be feasibly run on board using currently available hardware.

ACKNOWLEDGMENT

The authors would like to thank collaboration from the *Centre National d'Études Spatiales* (CNES) in the obtention of Pléiades simulated data used for the experiments.

REFERENCES

[1] (May 1, 2022). *UCS Satellite Database*. [Online]. Available: https://www.ucsusa.org/resources/satellite-database

[2] N. D. Beser, "Space data compression standards," *Johns Hopkins APL Tech. Dig.*, vol. 15, no. 3, pp. 206–223, 1994.

[3] B. Li, R. Yang, and H. Jiang, "Remote-sensing image compression using two-dimensional oriented wavelet transform," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 1, pp. 236–250, Jan. 2011.

[4] *Information Technology—Digital Compression and Coding of Continuous-Tone Still Images: Requirements and Guidelines*, International Organization for Standardization, Geneva, Switzerland, ISO/IEC Standard 10918-1, Feb. 1994.

[5] *Information Technology—JPEG 2000 Image Coding System—Part 1: Core Coding System*, ISO/IEC document Recommendation T.80, Dec. 2000.

[6] *Information Technology—High Efficiency Coding and Media Delivery in Heterogeneous Environments—Part 2: High Efficiency Video Coding*, International Organization for Standardization, Geneva, Switzerland, ISO/IEC Standard 23008-2, Aug. 2020.

[7] J. Ballé, V. Laparra, and E. Simoncelli, "End-to-end optimised image compression," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–27.

[8] L. Theis, W. Shi, A. Cunningham, and F. Huszár, "Lossy image compression with compressive autoencoders," in *Proc. Int. Conf. Learned Represent. (ICLR)*, Mar. 2017, pp. 1–19.

[9] J. Ballé, D. C. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–47.

[10] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Deep residual learning for image compression," in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshop*, Jun. 2019, pp. 1–5.

[11] J. Ballé et al., "Nonlinear transform coding," *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 2, pp. 339–353, Feb. 2021.

[12] Z. Cui, J. Wang, S. Gao, T. Guo, Y. Feng, and B. Bai, "Asymmetric gained deep image compression with continuous rate adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 10527–10536.

[13] J. Li and Z. Liu, "Multispectral transforms using convolution neural networks for remote sensing multispectral image compression," *Remote Sens.*, vol. 11, no. 7, p. 759, Mar. 2019.

[14] C. Deng, Y. Cen, and L. Zhang, "Learning-based hyperspectral imagery compression through generative neural networks," *Remote Sens.*, vol. 12, no. 21, p. 3657, Nov. 2020.

[15] V. A. de Oliveira et al., "Reduced-complexity end-to-end variational autoencoder for on board satellite image compression," *Remote Sens.*, vol. 13, no. 3, p. 447, Jan. 2021.

[16] J. Li and Z. Liu, "Efficient compression algorithm using learning networks for remote sensing images," *Appl. Soft Comput.*, vol. 100, Mar. 2021, Art. no. 106987.

[17] Y. Chong, L. Chen, and S. Pan, "End-to-end joint spectral–spatial compression and reconstruction of hyperspectral images using a 3D convolutional autoencoder," *J. Electron. Imag.*, vol. 30, no. 4, pp. 1–11, Mar. 2021.

[18] F. Kong, T. Cao, Y. Li, D. Li, and K. Hu, "Multi-scale spatial–spectral attention network for multispectral image compression based on variational autoencoder," *Signal Process.*, vol. 198, Sep. 2022, Art. no. 108589.

[19] Y. Guo, Y. Tao, Y. Chong, S. Pan, and M. Liu, "Edge-guided hyperspectral image compression with interactive dual attention," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5500817.

[20] V. A. de Oliveira et al., "Satellite image compression and denoising with neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[21] S. M. I. Verdú, J. Serra-Sagristà, J. Bartrina-Rapesta, M. Hernández-Cabronero, V. Laparra, and J. Ballé, "Hyperspectral image compression using convolutional neural networks with local spectral transforms and non-uniform sample normalisation," in *Proc. 8th Int. Workshop Onboard Payload Data Compress.*, 2022, pp. 1–8.

[22] Q. Xu et al., "Synthetic aperture radar image compression based on a variational autoencoder," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[23] Z. Di, X. Chen, Q. Wu, J. Shi, Q. Feng, and Y. Fan, "Learned compression framework with pyramidal features and quality enhancement for SAR images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[24] T. Dumas, A. Roumy, and C. Guillemot, "Autoencoder based image compression: Can the learning be quantization independent?" in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 1188–1192.

[25] Y. Choi, M. El-Khamy, and J. Lee, "Variable rate deep image compression with a conditional autoencoder," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3146–3154.

[26] F. Yang, L. Herranz, J. van de Weijer, J. A. I. Guitián, A. M. López, and M. G. Mozerov, "Variable rate deep image compression with modulated autoencoder," *IEEE Signal Process. Lett.*, vol. 27, pp. 331–335, 2020.

[27] F. Yang, L. Herranz, Y. Cheng, and M. G. Mozerov, "Slimmable compressive autoencoders for practical neural image compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4996–5005.

[28] S. Yin, C. Li, Y. Bao, Y. Liang, F. Meng, and W. Liu, "Universal efficient variable-rate neural image compression," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 2025–2029.

[29] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized Gaussian mixture likelihoods and attention modules," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7936–7945.

[30] D. Taubman and M. Marcellin, *JPEG2000 Image Compression Fundamentals, Standards and Practice*. New York, NY, USA: Springer, 2013.

[31] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, Dec. 2014, pp. 1–15.

[32] *Landsat 8 Google Cloud Data Portal*. Accessed: Jun. 6, 2021. [Online]. Available: Https://cloud.google.com/storage/docs/public-datasets/landsat

[33] Jet Propulsion Laboratory. *AVIRIS Data Portal*. NASA. Accessed: Nov. 6, 2021. [Online]. Available: https://aviris.jpl.nasa.gov/dataportal/