Low Bit Rate 2D Seismic Image Compression With Deep Autoencoders

Ana Paula Schiavon[1], João Paulo Navarro[2], Marcelo Bernardes Vieira[1], and Pedro Mário Cruz e Silva[2]*

[1] Universidade Federal de Juiz de Fora, Juiz de Fora, Brasil
apschiavon@ice.ufjf.br, marcelo.bernardes@ufjf.edu.br
[2] NVIDIA, São Paulo, Brasil
{jpnavarro,pcruzesilva}@nvidia.com

Abstract. In this paper, we present a deep learning approach for very low bit rate seismic data compression. Our goal is to preserve perceptual and numerical aspects of the seismic signal whilst achieving high compression rates. The trade-off between bit rate and distortion is controlled by adjusting the loss function. 2D slices extracted from seismic 3D amplitude volumes feed the network for training two simultaneous networks, an autoencoder for latent space representation, and a probabilistic model for entropy estimation. The method benefits from the intrinsic characteristic of deep learning methods and automatically captures the most relevant features of seismic data. An approach for training different seismic surveys is also presented. To validate the method, we performed experiments in real seismic datasets, showing that the autoencoders can successfully yield compression rates up to 68:1 with an average PSNR around 40 dB.

Keywords: Seismic Data Compression · Deep Autoencoders · Geophysical Image Processing · High Bit-Depth Compression.

1 Introduction

The quality of acquisition sensors has been evolved significantly in the past years. This fact implies on higher resolution signals to process, to storage, and to transmit. The use of effective compressing algorithms plays an important role in seismic processing, aiming to deal with the substantial increase in data resolution. Generally speaking, reliance on compression algorithms in terms of signal reconstruction is a concern in the field due to the dilemma of choosing *lossless* methods, with perfect reconstruction, or *lossy* compression, with a greater reduction on storage with allowed reconstruction distortions.

Typical compression methods benefit from the extensive oscillatory nature of the seismic data to model the algorithms. This leads to approaches involving

* Authors thank CAPES, FAPEMIG (grant CEX-APQ-01744-15) for the financial support, and NVIDIA for the donation of one GPU as part of the GPU Grant Program.

transformations to wavelets and cosine domains [1]. The so-called transform-based methods consider the representation of the volume in these domains, storing and processing only on a small subset of the total coefficients. A wavelet-transform based compression algorithm was proposed in [2], allowing extremely high compression ratios with considerably small errors. Other techniques such as low-rank methods are focused in working directly on lower dimensional matrices sampled from the higher dimensional wavefield [3].

Recently, some methods explored standard video and image compression techniques. With similar performance to a licensed commercial wavelet-based scheme used by the industry, experiments performed in [4] indicated that the JPEG-XR can be used to fast compress seismic images allowing to control the quality or bit rate target. Motivated by the performance of video codecs, a codec under the HEVC [5] intra coding framework is presented in [6] to compress seismic images, outperforming the previously published compression schemes. Considering the similarity between 3D seismic data and videos, an extension of this method was proposed in [7] aiming to explore the temporal redundancy in three-dimensional data. This method surpasses the previous method and it is less time-consuming.

However, one may notice that most of these approaches are exposed to any sort of dataset bias. The challenge in working with compression for seismic domain relies on the difficulty of detaching parts of the signal that represent physical properties from those who do not. We argue that capturing all variances and inconsistencies that may be present in seismic signals such as noise, interferences, and processing inaccuracies in a deterministic fashion is not practical. Machine Learning techniques is a viable way to face this problem. In this sense, these techniques were employed in [8] to compress seismic signals directly from the field. They trained a shallow autoencoder to compress the data while a Restricted Boltzmann Machine was used to optimize its parameters. In addition to achieving interesting preliminary results (10:1 and a PSNR of 30dB), this approach can capture the most representative features using only a portion of the dataset.

The popularity of Deep Learning (DL) algorithms has considerably increased in recent years due to consistent advances in solving complex computer science tasks such as image classification, speech to text, and translation. Recently, deep neural network approaches are taking momentum in solving low-level problems in image processing, such as super-resolution and image compression, leading to impressive state-of-the-art results. Complementing the autoencoder with an adversarial training, the proposal of [9] to image compression outperforms all previous codecs producing visually agreeable reconstructions for very low bit rates. With competitive performance to the previous work, [10] proposed an image compression system based in two networks trained concurrently. A probabilistic model is used to learn the dependencies between symbols in the autoencoder latent representation, and another autoencoder uses it for entropy estimation, in order to control the rate-distortion trade-off.

The main contribution of this work is the extension of the method proposed by [10] for low bit rate compression of seismic volumes. More specifically, we propose a training and inference schemes tuned for seismic compression of multiple volumes. Our method is trained using a collection of seismic 2D-slices to feed the network. A DL method based on conditional probabilistic deep autoencoder is tuned to exploit the inherent features of seismic data.

2 Proposed method

We extend the end-to-end pipeline proposed by [10]. Our goal is to compress an amplitude seismic volume training simultaneously two deep networks: a compressing autoencoder and a probabilistic model. The first deals with the rate-distortion trade-off between a small number of bits and small distortions. The second is a 3D-CNN that learns the dependencies between the symbols of the autoencoder latent representation. Both models were trained to balance the trade-off. We provide training and inference schemes for 2D-slice based seismic volume compression.

2.1 Probabilistic autoencoder

By definition, an autoencoder is an unsupervised learning algorithm that is trained to adjust its weights aiming to set the target values to be equal to the inputs. The compressive autoencoder is a model composed of an encoder, a decoder, and a quantizer [11]. The encoder $E : \mathcal{R}^d \rightarrow \mathcal{R}^m$ maps the input $\mathbf{x}$ to a lower dimension latent space. The quantizer $Q : \mathcal{R} \rightarrow C$ discretizes the latent representation $\mathbf{z}$, obtaining $\hat{\mathbf{z}} = Q(\mathbf{z})$, and allowing it to be losslessly encoded into a bitstream through arithmetic coding strategy. The decoder D reconstructs the image $\hat{\mathbf{x}} = D(\hat{\mathbf{z}})$ from its quantized representation through the losslessly decoded bitstream. The goal is to minimize the rate-distortion trade-off $d(\mathbf{x}, \hat{\mathbf{x}}) + \beta H(\hat{\mathbf{z}})$, where d is a function that measures the distortion between the original image and its reconstruction, H is the entropy of the quantized latent representation and β controls the trade-off.

The quantization combines the works of [12] and [13]. The authors used a clustering based quantization, where each entry of the latent representation is changed by the index of the nearest centroid. The encoder and the decoder are 2D-CNNs with the particularity that the last layer of the encoder has an additional feature map, named importance map. This map is used to generate a binary 3D mask $\lceil \mathbf{m} \rceil$ that is applied to the feature maps volume. It allows different regions of the image to be represented by a different number of bits, according to the detail level. Since the mask binarization is not a differentiable operation, a soft approximation in the backward pass of the back propagation is used.

The probabilistic model $P(\hat{\mathbf{z}})$ is a 3D-CNN that models the conditional probability of a symbol belonging to a centroid given the previous symbols to it, learning their dependencies in the latent representation of the autoencoder [14]. Considering the distribution

$$p(\hat{\mathbf{z}}) = \prod_{i}^{m} (\hat{\mathbf{z}}_i | \hat{\mathbf{z}}_{i-1}, \ldots, \hat{\mathbf{z}}_1), \tag{1}$$

this network is used to estimate each term $(\hat{\mathbf{z}}_i | \hat{\mathbf{z}}_{i-1}, \ldots, \hat{\mathbf{z}}_1)$:

$$P_{i,l}(\hat{\mathbf{z}}) \approx p(\hat{\mathbf{z}}_i = C_l | \hat{\mathbf{z}}_{i-1}, \ldots, \hat{\mathbf{z}}_1), \tag{2}$$

where $l = \{1, \ldots, L\}$ and L is the size of the centroids set C.

The losses of both models are based on the rate-distortion trade-off. Given the set of training seismic images $\mathbf{X}$, we train over minibatches $\mathbf{X}_B = \left\{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(B)}\right\}$ of crops from $\mathbf{X}$.

Since the models are trained simultaneously, the entropy term H is calculated using two approaches: first, instead of encoding the entire masked symbol volume, the 3D binarized mask $\lceil \mathbf{m} \rceil$ is encoded and subsequently the symbols of the volume that are not zero. This is used by the autoencoder, since it allows the encoder to easily control the spatial allocation of bits [15]. The second, uses the distribution $p(\hat{\mathbf{z}})$, since it does not have direct access to the mask and needs to learn the dependencies on the entire masked symbol volume [14]. The loss function for the probabilistic model P is defined as:

$$\mathcal{L}_P := \frac{1}{B} \sum_{j=1}^{B} d(\mathbf{x}^{(j)}, \hat{\mathbf{x}}^{(j)}) + \beta \sum_{i=1}^{m} - \log P_{i,I(\hat{z}_i^{(j)})}, \tag{3}$$

where $P_{i,I(\hat{z}_i^{(j)})}$ specifies for each voxel i in the entire masked symbol volume the probabilities of belonging to each index $I(\hat{z}_i)$ of the centroids set. The loss function for the autoencoder is given by:

$$\mathcal{L}_{E,D,Q} = \frac{1}{B} \sum_{j=1}^{B} d(\mathbf{x}^{(j)}, \hat{\mathbf{x}}^{(j)}) + \beta \sum_{i=1}^{m} - \lceil \mathbf{m}_i \rceil \log P_{i,I(\hat{z}_i^{(j)})}. \tag{4}$$

Notice that this loss incorporates the probabilistic model as the entropy term of the autoencoder with the benefit of being weighted by $\lceil \mathbf{m} \rceil$. In next sections, we propose training and inference schemes, based on this compression model, taking into account the characteristics of seismic 2D-slices.

2.2 Training Scheme

Fig. 1 presents our training pipeline. Initially, we perform a preprocessing step, aiming to adapt the volumes to the network input. Seismic sections are numerically represented as one channel 32-bit floating-point. But the model proposed by [10] was designed for general purpose image compression with 3-channels of 8-bit unsigned integers. We have verified that pre-trained models based on ImageNet did not fit our specific data domain. The fine-tuning approach using initial

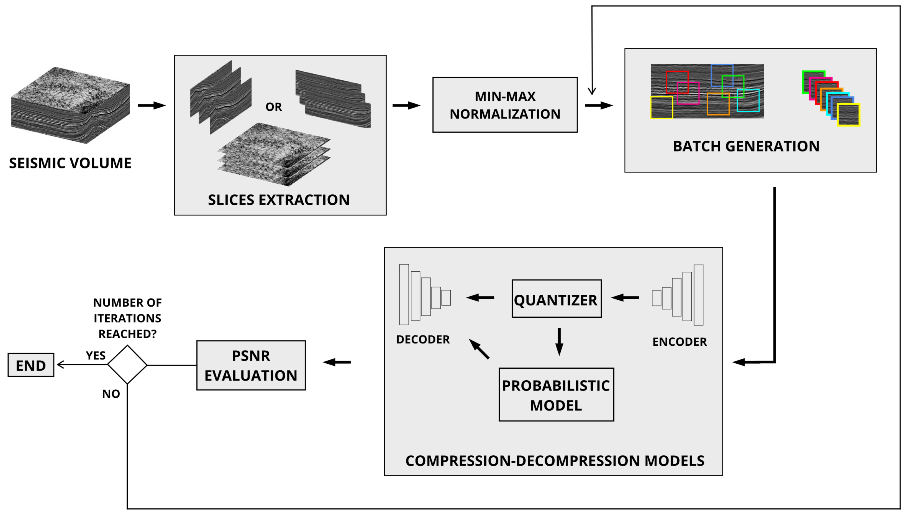


Fig. 1: Overview of the proposed training scheme.

weights from 8-bit images for network optimization led to higher iterations until convergence. For this reason, we adapted the network and all metrics to work with seismic data and trained all models from scratch for a collection of volumes.

The encoder-decoder performs 2D convolutions and thus the training dataset was formed by extracting slices of the inline, crossline or time-depth directions. Then, the slices are normalized using min-max strategy. This is important because the original network was designed to work in the [0,1] interval quantized with 8-bits for low-dynamic range images. Since seismic data are quantized with 32-bits, its range values are wider and the min-max is arbitrary across different volumes. In order to train multiple volumes at the same time, the min-max from all volumes are used.

The batch generation is performed extracting random crops from the slices and randomly flipping them. In the case of a training set composed by various datasets, the batch is built with crops from all of them at the same quantity. In this way, we are preventing the model from being biased by one dataset. Since the number of slices of a dataset can vary, smaller datasets will provide crops of repeated slices. In general, given the random nature of the cropping, the chance of an exactly repeated crop is negligible.

The batch is then used to feed the encoder. The output of the encoder is quantized and it is used by the probabilistic model to estimate the entropy and to calculate the centroids. These centroids and the quantizer output are used by the decoder to reconstruct the batch images. We perform a PSNR evaluation between the initial and reconstructed batches. This metric was chosen to be maximized by the training step instead the original Multi-Scale Structural Similarity (MS-SSIM) for images. The PSNR is simpler to compute and yielded better overall results. The training step is repeated until the maximum number of iterations is reached.

2.3 Inference Scheme

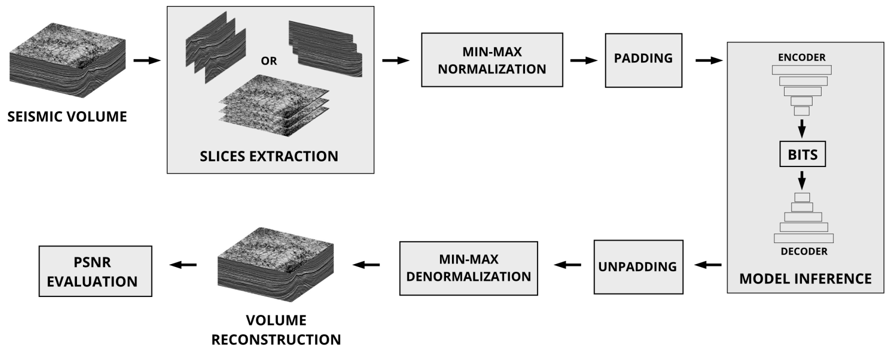


Fig. 2: Overview of the proposed inference scheme.

The inference of the DL model is performed according to the Fig. 2. After the training step, the network is tuned for compression of seismic slices normalized in [0,1] range. Given a seismic volume, the slices are extracted in one of the directions and then normalized with its min-max values. If their shapes are not divisible by the network subsampling factor, they are padded with a border extension. We propose the symmetric border extension since it better preserves the frequencies of the seismic volume. The encode/decode are then performed for model inference. Both input and output are unpadded to guarantee coherence of the metric evaluation. The slices are denormalized to reconstruct the compressed seismic volume. Finally, we evaluate the error between the original and reconstructed volumes.

3 Experimental results

To validate the proposed method, we perform experiments in different 3D stacked seismic volumes available on SEG Open Data repository [16]. Our method was implemented using the TensorFlow framework, and all runs performed on a single GPU NVIDIA Tesla V100. We verified that the following hyper-parameter space was suitable for all datasets: Adam optimizer, batch-size of 32, initial learning rate of $8 \cdot 10^{-5}$ for the autoencoder and $1 \cdot 10^{-4}$ for the probabilistic model, both with step decay of 0.1 every 10 epochs, and crop size of 128×128. Reconstruction quality is reported as PSNR and SNR in decibels (dB) due to its sensibility to small error variations, and compression rate as bits-per-voxel (bpv), expressing the average number of bits necessary to represent the 32-bits amplitude values.

3.1 Training Protocol

From SEG Open Data repository [16] we selected five seismic surveys: Kahu3D, Parihaka3D, Netherlands F3-Block, Penobscot3D and Waihapa3D volumes. Netherlands F3-Block, Penobscot3D and Waihapa3D were used only for testing. We also reserved 10% of training volume slices for validation (randomly chosen), so we could define appropriate hyper-parameters such as number of epochs. Normalizing by standardization leads to impressive reduction in epochs, with initial average PSNR of 25 dB in contrast with 10 dB.

The details of the seismic surveys, such as size and grid dimension are shown in Table 1. Slices were extracted from volumes in (x, z) (inline), (y, z) (crossline) and (x, y) (time-depth) planes.

Table 1: Uncompressed dataset properties.

Dataset	Size (GB)	Grid Dimension
Kahu3D	6.17	$(584 \times 1695 \times 1498)$
Parihaka3D	3.86	$(920 \times 1124 \times 874)$
Netherlands F3-Block	1.25	$(631 \times 951 \times 463)$
Penobscot3D	0.60	$(401 \times 301 \times 1251)$
Waihapa3D	0.29	$(201 \times 291 \times 1238)$

3.2 Results and Discussion

Due its similarity, we use the inline and crossline directions alongside to train a single model with good performance in both directions. Since the time-depth direction is too different from the previous, its results were obtained training another model using only this direction.

Table 2 shows the comparison between the three possible planes used in the inference step. Notice that the crossline direction has similar results compared to the inline. This is expected since the volume in both directions have similar characteristics. The results of the model trained and inferred with the time-depth plane is clearly worse than the others. The data in the time-depth directions seems noisy if compared to the other planes. The compression model is less efficient, requiring an extra tuning of its hyper-parameters.

We trained the model using only slices from Kahu and Parihaka datasets. Netherlands F3-Block is known to be a noisy survey, thus naturally harder to compress. But even under this condition, it took only four epochs for the model to reach an interesting PSNR of 35.56. This result highlights the generalization capabilities of the method. With a SNR = 29.56, however, one may not consider that the relevant original information was preserved. A deeper qualitative analysis is needed to correlate the minimum bit rate and PSNR that do not result

Table 2: Results for testing sets. Overall average compression ratio is 68:1.

Dataset	Direction	*bpv*	PSNR	SNR
Netherlands F3_Block	Inline	0.45	35.56	29.56
	Crossline	0.46	35.07	29.08
	Time-depth	0.40	33.45	27.46
Penobscot3D	Inline	0.39	40.48	34.47
	Crossline	0.37	42.00	35.99
	Time-depth	0.36	37.11	31.10
Waihapa3D	Inline	0.65	34.81	29.30
	Crossline	0.66	34.55	29.04
	Time-depth	0.47	35.21	29.70

in relevant losses. We also need to train with more seismic surveys to further increase generality power. Since the software and the data used by the state-of-the-art results are private, we are not able to report the performance of our method on them.

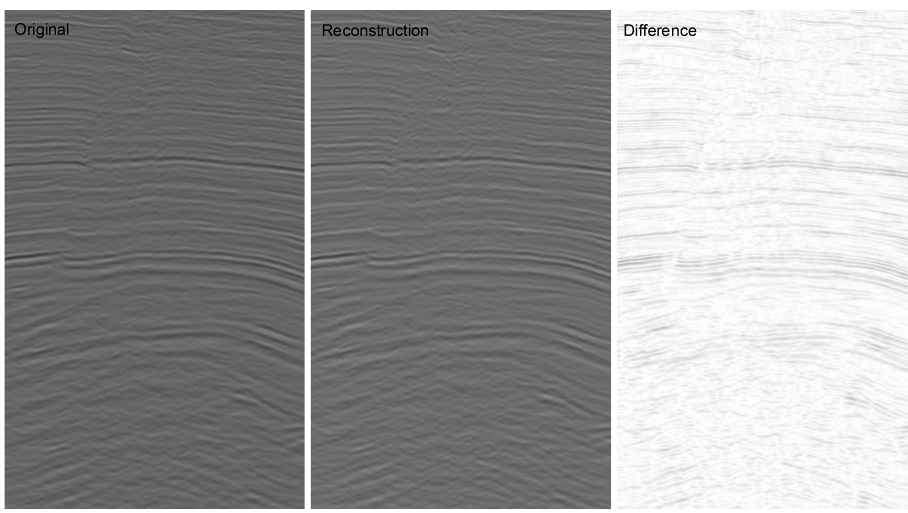
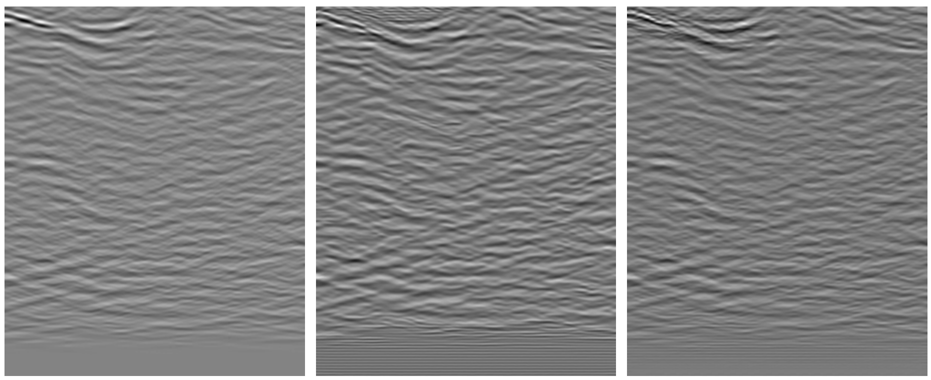


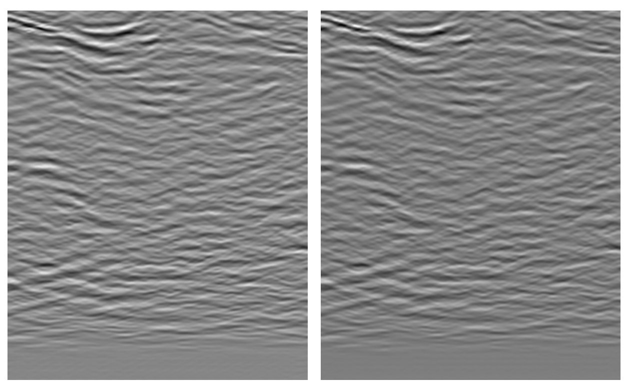
Fig. 3: Section reconstruction from an image in Penobscot test set. Despite the minor differences (right), overall reconstruction preserves most relevant seismic features, important for geological interpretations. Compression ratio = 86:1 and PSNR = 42 dB.

Figure 3 depicts a complete slice compression and decompression result for the Penobscot dataset. Even with a 86:1 compression, the details are fairly preserved with PSNR = 42 dB. Figure 4 shows a crop from a slice of the Waihapa volume (Fig. 4a) compressed in different bit rates. The extreme compression in

Figure 4b and 4c introduced high frequencies in the original slice (Fig. 4a). One may notice a less effective PSNR gain as we move to higher compression ratios when compared to HVEC-based methods [6, 7]. Considering the Figures 3, 4d and 4e, along with our observations, bit rates resulting in at least PSNR = 36 dB tend to better preserve details.



(a) Original (b) 0.227 bpv, 31.54 dB (c) 0.657 bpv, 34.55 dB



(d) 0.841 bpv, 36.33 dB (e) 1.385 bpv, 38.46 dB

Fig. 4: Two dimensional crops extracted from Waihapa volume. The method maximizes PSNR during optimization while achieving low bit rates.

As aforementioned, we performed the standardization operation on both first and last layers, leading to a drastically reduced convergence time. Using a single NVIDIA Tesla V100 we were able to train a model within the interval of 1 hour. We expect to explore parallel approaches such as Horovod [17] to distribute training between multi-GPU as we increase total amount of training data.

4 Conclusions

In this paper, we presented a deep learning approach for 3D seismic data compression. The network is fed with two-dimensional 32-bits sections from the original volume, training at the same time two networks, one for latent space representation, and other for entropy estimation. The bit rate of the compressed volume is controlled by hyper-parameter tuning. The method benefits from the intrinsic characteristic of deep learning approaches and automatically captures the most relevant features of the data. It presents promising results in data reconstruction with high PSNR values at low bit rates. High compression rates (up to 68:1 and PSNR = 40 dB in average) are obtained by training models with multiple seismic surveys on a single procedure. As future work, we intend to improve our workflow so more volumes are aggregated to the training dataset without introducing biases.

References

1. Averbuch, A.Z., Meyer, F., Stromberg, J.., Coifman, R., Vassiliou, A.: Low bit-rate efficient compression for seismic data. IEEE Transactions on Image Processing **10**(12), 1801–1814 (Dec 2001). https://doi.org/10.1109/83.974565
2. Villasenor, J. D. and Ergas, R. A. and Donoho, P. L.: Seismic data compression using high-dimensional wavelet transforms. In: Proceedings of Data Compression Conference-DCC'96, pp. 396–405, IEEE, (1996)
3. Zhang, Y., Da Silva, C., Kumar, R., Herrmann, F., et al.: Massive 3d seismic data compression and inversion with hierarchical tucker. In: 2017 SEG International Exposition and Annual Meeting. Society of Exploration Geophysicists (2017)
4. Liu, Y., Xiong, Z., Lu, L., Hohl, D.: Fast snr and rate control for jpeg xr. In: Signal Processing and Communication Systems (ICSPCS), 2016 10th International Conference on. pp. 1–7. IEEE (2016)
5. Sullivan, G.J., Ohm, J.R., Han, W.J., Wiegand, T., et al.: Overview of the high efficiency video coding(hevc) standard. IEEE Transactions on circuits and systems for video technology **22**(12), 1649–1668 (2012)
6. Radosavljević, M., Xiong, Z., Lu, L., Vukobratović, D.: High bit-depth image compression with application to seismic data. In: Visual Communications and Image Processing (VCIP), 2016. pp. 1–4. IEEE (2016)
7. Radosavljević, M., Xiong, Z., Lu, L., Hohl, D., Vukobratović, D.: Hevc-based compression of high bit-depth 3d seismic data. In: Image Processing (ICIP), 2017 IEEE International Conference on. pp. 4028–4032. IEEE (2017)
8. Nuha, H., Mohandes, M., Liu, B.: Seismic-data compression using autoassociative neural network and restricted boltzmann machine. SEG Technical Program Expanded Abstracts 2018 pp. 186–190 (2018)
9. Rippel, O., Bourdev, L.: Real-time adaptive image compression. In: International Conference on Machine Learning. pp. 2922–2930 (2017)
10. Mentzer, F., Agustsson, E., Tschannen, M., Timofte, R., Van Gool, L.: Conditional probability models for deep image compression. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
11. Theis, L., Shi, W., Cunningham, A., Huszár, F.: Lossy image compression with compressive autoencoders. CoRR **abs/1703.00395** (2017)

12. Agustsson, E., Mentzer, F., Tschannen, M., Cavigelli, L., Timofte, R., Benini, L., Gool, L.V.: Soft-to-hard vector quantization for end-to-end learning compressible representations. In: Advances in Neural Information Processing Systems. pp. 1141–1151 (2017)
13. Theis, L., Shi, W., Cunningham, A., Huszár, F.: Lossy image compression with compressive autoencoders. In: International Conference on Learning Representations (2017), https://openreview.net/pdf?id=rJiNwv9gg
14. Van Oord, A., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. In: International Conference on Machine Learning. pp. 1747–1756 (2016)
15. Li, M., Zuo, W., Gu, S., Zhao, D., Zhang, D.: Learning convolutional networks for content-weighted image compression. arXiv preprint arXiv:1703.10553 (2017)
16. SEG: Open data. http://wiki.seg.org/wiki/Open_data (2019), accessed: 2019-01-10
17. Sergeev, A., Balso, M.D.: Horovod: fast and easy distributed deep learning in tensorflow. CoRR **abs/1802.05799** (2018), http://arxiv.org/abs/1802.05799