

DRASIC: Distributed Recurrent Autoencoder for Scalable Image Compression

Enmao Diao*, Jie Ding[†], and Vahid Tarokh*

*Duke University [†]University of Minnesota-Twin Cities
 Durham, NC, 27701, USA Minneapolis, MN 55455, USA
 enmao.diao@duke.edu dingj@umn.edu
 vahid.tarokh@duke.edu

Abstract

We propose a new architecture for distributed image compression from a group of distributed data sources. The work is motivated by practical needs of data-driven codec design, low power consumption, robustness, and data privacy. The proposed architecture, which we refer to as Distributed Recurrent Autoencoder for Scalable Image Compression (DRASIC), is able to train distributed encoders and one joint decoder on correlated data sources. Its compression capability is much better than the method of training codecs separately. Meanwhile, the performance of our distributed system with 10 distributed sources is only within 2 dB peak signal-to-noise ratio (PSNR) of the performance of a single codec trained with all data sources. We experiment distributed sources with different correlations and show how our data-driven methodology well matches the Slepian-Wolf Theorem in Distributed Source Coding (DSC). To the best of our knowledge, this is the first data-driven DSC framework for general distributed code design with deep learning.

1 Introduction

It has been shown by a variety of previous works that deep neural networks (DNN) can achieve comparable results as classical image compression techniques [1–9]. Most of these methods are based on autoencoder networks and quantization of bottleneck representations. These models usually rely on entropy codec to further compress codes. Moreover, to achieve different compression rates it is unavoidable to train multiple models with different regularization parameters separately, which is often computationally intensive.

In this work, we are motivated to develop an architecture that has the following advantages. First, unlike classical distributed source coding (DSC) which requires customized code design for different scenarios [10], a data-driven distributed compression framework can handle nontrivial distribution of image sources with arbitrary correlations. Second, the computation complexity of encoders (e.g. mobile devices) can be transferred to the decoder (e.g. a remote server). Such a system of low complexity encoders can be used in a variety of application domains, such as multi-view video coding [11], sensor networks [10], and under-water image processing where communication bandwidth and computational power are quite restricted [12, 13]. Third, the distributed framework can be more robust against heterogeneous noises or malfunctions of encoders, and such robustness can be crucial in,

This work was supported in part by Office of Naval Research Grant No. N00014-18-1-2244. We provide our implementation at <https://github.com/dem123456789/Distributed-Recurrent-Autoencoder-for-Scalable-Image-Compression>

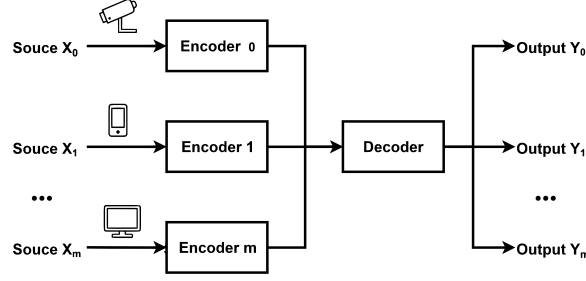


Figure 1: Illustration of Deep Distributed Source Coding.

e.g., unreliable sensor networks [11, 14, 15]. Last but not least, the architecture is naturally scalable in the sense that codes can be decoded at more than one compression quality level, and it allows efficient coding of correlated sources which are not physically co-located. This is especially attractive in video streaming applications [16, 17].

It is tempting to think that splitting raw data for different encoders compromises the compression quality. It is thus natural to ask this question: Can distributed encoders perform as well as a single encoder trained with all data sources together? A positive answer from a theoretical perspective was given in the context of information theory, where DSC is an important problem regarding the compression of multiple correlated data sources. The Slepian-Wolf Theorem shows that lossless coding of two or more correlated data sources with separate encoders and a joint decoder can compress data as efficiently as the optimal coding using a joint encoder and decoder [18, 19]. The extension to lossy compression with Gaussian data sources was proposed as Wyner-Ziv Theorem [20]. Although these theorems were published in 1970s, it was after about 30 years that practical applications such as Distributed Source Coding Using Syndromes (DISCUS) emerged [21]. One of the main advantages of DSC is that the computation complexity of the encoder is transferred to the decoder. A system architecture with low complexity encoders can be a significant advantage in applications such as multi-view video coding and sensor networks [10, 11].

Motivated by the theoretical development of DSC, in this work we propose a DNN architecture that consists of distributed encoders and a joint decoder (illustrated in Fig. 1 and 2). We show that distributed encoders can perform as well as a single encoder trained with all data sources together. Our proposed DSC framework is data-driven by nature, and it can be applied to distributed data even with unknown correlation structure.

The paper is outlined below. We review previous related works in Section 2. We describe our proposed architecture for general image compression and its basic modules in Subsections 3.1-3.4. Then we elaborate the Deep Distributed Source Coding framework in Subsection 3.5. Experimental results are shown in Section 4, followed by conclusions in Section 5.

2 Related Work

Though there has been a variety of research on lossy data compression in the past few decades, little attention has been paid to a systematic approach for general and practical distributed code design, especially in the presence of an arbitrary number of nontrivial data

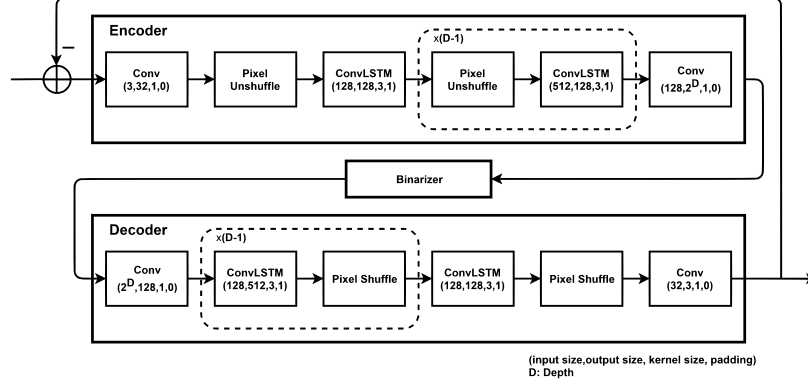


Figure 2: Illustration of Recurrent Autoencoder for Scalable Image Compression.

sources with arbitrary correlations [10]. A main motivation of this work is to attempt to replace the practical hand-crafted code design with data-driven approaches. To our best knowledge, what we propose is the first data-driven DSC architecture. Unlike hand-crafted quantizers, our neural network-based quantizers show that the correlations among different data sources can be exploited by the model parameters. Inspired by DSC, We empirically show that it is possible to approach the theoretical limit with our methodology.

2.1 Image compression with Deep Learning

There exist a variety of classical codecs for lossy image compression. Although the JPEG standard [22] was developed thirty years ago, it is still the most widely used image compression method. Several extensions to JPEG including JPEG2000 [23], WebP [24] and BPG [25] have been developed. Most of these classical codecs rely on a quantization matrix applied to the coefficients of discrete cosine transform or wavelet transform.

Common deep neural network architecture for image compression are auto-encoders including non-recurrent autoencoders [2, 5, 8, 9] and recurrent autoencoders [1, 4, 6]. Non-recurrent autoencoders use entropy codec to encode quantized bottleneck representations, and recurrent models introduce incremental binarized codes at each compression quality. The generated codes of non-recurrent models is not scalable and their performance heavily relies on the conditional generative model like PixelCNN [26] which arithmetic coding can take advantage of [8, 9]. Recurrent autoencoders, on the other hand, can reconstruct images at lower compression qualities with the subset of high quality codes. Other notable variations include adversarial training [27], multi-scale image compression [28], and generalized divisive normalization (GDN) layers [2]. Another challenge is to well define the derivative of quantizations of bottleneck representations. [2] replaced non-differentiable quantization step with a continuous relaxation by adding uniform noises. [1], on the other hand, used a stochastic form of binarization.

2.2 Distributed Source Coding

Our methodology is inspired by the information-theoretic results on DSC which have been established since 1970s. The Slepian-Wolf [18] Theorem shows that two correlated data

sources encoded separately and decoded jointly can perform as well as joint encoding and decoding, and outperform separate encoding and separate decoding. The striking result indicates that as long as the codes are jointly decoded, there can be no loss in coding efficiency even the codes are separately encoded. Cover [19] generalizes the achievability of Slepian-Wolf coding to arbitrary number of correlated sources. [20] Coding gives a rate-distortion curve as an extension to lossy cases. Some researchers have also shown the applicability of DSC on still images [29]. In practical applications, low complexity video encoding benefits from the DSC framework which can transfer the complexity of encoder to decoder [30,31]. Scalable Video Coding can also be incorporated with DSC [32]. These proposed methods indicate the feasibility of DSC in our problem setting.

3 Methods

In this section, we first describe the recurrent autoencoder for scalable image compression used in our work. We will then describe how this Deep Learning architecture is used in Distributed Source Coding framework.

3.1 Network Architecture

Our compression network consists of an encoder, a binarizer, and a decoder. The activation function following each Convolutional Neural Network (CNN) module is \tanh . For the first iteration of our model, the input images are initially encoded and transformed into $(-1, 1)$ by \tanh activation function. Binary codes are quantized from bottleneck representations. The decoder then reconstructs images based on the received binary codes. Finally, we compute the residual difference between the original input images and the reconstructed output images. At the next iteration, the residual difference is feedback as the new input for our model. This procedure is repeated multiple iterations to gain more codes for better reconstruction performance. Therefore, the reconstructed images at each iteration are the sum of output reconstructions from previous and current iterations. The dependencies among iterations are modeled by recurrent models like ConvLSTM. We iterate 16 times to generate scalable codes. Compared to non-scalable codes which require new set of codes at each compression quality, scalable codes are able to reconstruct images at lower compression quality by using the subset of codes. This is especially attractive in video streaming applications [16, 17].

Consider dataset $X = \{x\}^N$ consisting of N i.i.d. samples of some continuous or discrete variables x . The data generating process is unknown. Autoencoders for compression and reconstruction can be formulated in the following way. Data can be compressed with a neural network-based encoder $f(x; \theta)$ into quantized codes \tilde{z} and reconstructed with a decoder $g(\tilde{z}; \phi)$. We can binarize bottleneck representations z and control the compression quality by varying its channel sizes. The loss function $\mathcal{L}(x, \tilde{x})$ is minimized with respect to the model parameters θ and ϕ .

$$z = f(x; \theta), \tilde{z} = \text{Binarize}(z), \tilde{x} = g(\tilde{z}; \phi), \quad (1)$$

$$\text{Minimize } \mathcal{L}(x, \tilde{x}) \quad (2)$$

Deep recurrent autoencoder gradually increases compression quality by creating a correlated residual sequence from the difference between the input and output of our model. The advantage of recurrent model is that we can use a subset of generated codes to reconstruct images at lower compression qualities. Classical autoencoders, on the contrary, not only have to train multiple networks with different penalty coefficients for rate-distortion loss but also have to generate different codes for different compression quality. Suppose T iterations are used, we can formulate the recurrent autoencoder in the following way.

$$z_t = f(x_t; \theta), \tilde{z}_t = \text{Binarize}(z_t), \quad (3)$$

$$\tilde{x}_t = g(\tilde{z}_t; \phi), x_{t+1} = x_t - \tilde{x}_t, \tilde{x}_1 = 0, \quad (4)$$

$$\text{Minimize } \frac{1}{T} \sum_{t=1}^T \mathcal{L}(x_1, \sum_{i=1}^t \tilde{x}_i). \quad (5)$$

3.2 Deep Distributed Source Coding Framework

Fig. 1 and 2 illustrate our Distributed Recurrent Autoencoder for Scalable Image Compression (DRASIC). Similar to classical DSC framework, each data source is encoded separately and decoded jointly. In our network, each distributed encoder in Fig. 1 has the exact same structure in Fig. 2. Traditionally, researchers have to design different kind of codes for specific data sources [33]. We propose to use data-driven approach to handle complex scenarios where the distribution of data sources is unknown and their correlations can be arbitrary. Our proposal may also shed new light on sophisticated application scenarios such as videos where data sources and correlations are time dependent.

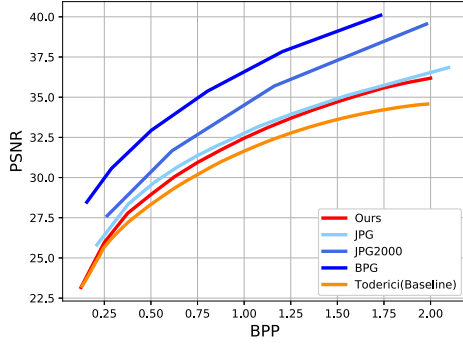
In our neural network-based DSC, M distributed encoders encode corresponding data sources x^m that can be arbitrarily correlated. Each neural network-based encoder $f(x^m; \theta^m)$ has their own model parameters θ^m . After binarizing bottleneck representations z^m , code sources \tilde{z}^m are transmitted and concatenated batch-wisely. A single decoder $g(\tilde{z}^m; \phi)$ reconstructs images \tilde{x}^m from all sources with the same model parameters ϕ . In classical settings, the joint decoder has to process all compressed codes from each source jointly. In our data-driven setting, the joint training process optimizes the model such that the single decoder can decode from correlated sources. In this case, decoding codes from a particular data source does not depend on synchronization of codes from other sources, since the model has been optimized to adapt the correlations among all sources.

$$z_t^m = f(x_t^m; \theta^m), \tilde{z}_t^m = \text{Binarize}(z_t^m), \quad (6)$$

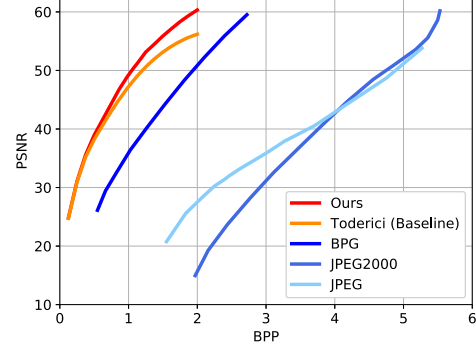
$$\tilde{x}_t^m = g(\tilde{z}_t^m; \phi), x_{t+1}^m = x_t^m - \tilde{x}_t^m, \tilde{x}_1^m = 0, \quad (7)$$

$$\text{Minimize } \frac{1}{MT} \sum_{t=1}^T \sum_{m=1}^M \mathcal{L}(x_1^m, \sum_{i=1}^t \tilde{x}_i^m). \quad (8)$$

Our result shows that the resulting distributed model can perform as well as encoding all data by one single encoder. However, if we encode and decode each data source separately, the performance becomes significantly worse, i.e. with $\tilde{x}_t^m = g(\tilde{z}_t^m; \phi^m)$.

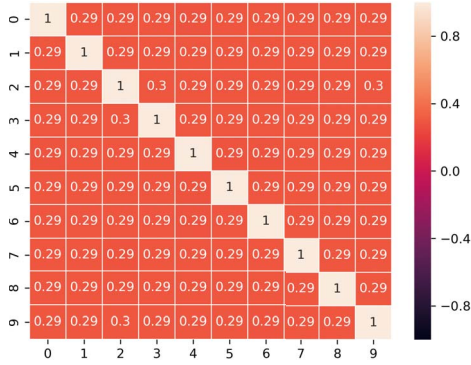


(a) PSNR vs. BPP on Kodak dataset

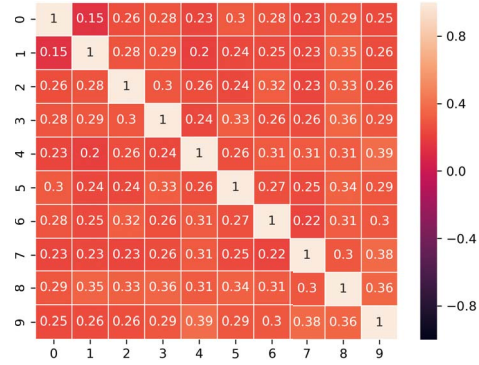


(b) PSNR vs. BPP on MNIST dataset

Figure 3: Our symmetric recurrent autoencoder performs comparable to classical codecs and neural network-based codecs.



(a) Split by random subsets.



(b) Split by class labels.

Figure 4: Pearson's correlation matrix among MNIST dataset.

4 Experiments

To show our model is capable of compressing natural images, we train our model on CIFAR10 dataset [12] and evaluate the rate-distortion curve on Kodak dataset [34]. To show our model is capable of compressing grayscale images and demonstrate the feasibility of training encoders in a distributed manner, we train and evaluate our models with MNIST dataset [35]. We observe that many non-recurrent autoencoders outperform recurrent models on rate-distortion curves [8, 9]. We emphasize the distinction between the recurrent and non-recurrent autoencoders which do not have the scalability of reconstructing low quality images by using the subset of codes for high quality reconstruction. Our experiments aim to empirically demonstrate the feasibility of scalable distributed source coding in a data-driven setting. We use Adam optimizer [36] with minibatch size of 100 for all experiments. We use learning rate 0.001 for a total of 200 epochs and decay every 50 epochs by a factor of 0.5. Fig. 3a shows that our symmetric recurrent autoencoder performs comparable to classical codecs and neural network-based codecs on compressing natural images, and performs significantly better on compressing handwritten grayscale images.

To demonstrate the feasibility of compressing distributed data sources, we split our data into correlated subsets to emulate the case where encoders only have access to distributed correlated data sources. We conduct our experiments with (2, 4, 8, 10) number of

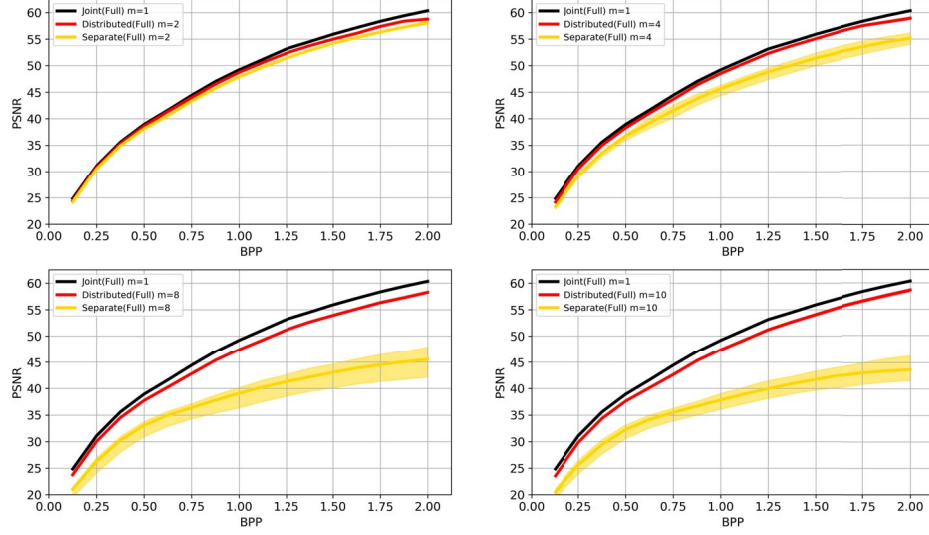


Figure 5: Rate-distortion curves for data sources distributed by random subsets with $T = 16$ for all sources.

distributed sources. For the MNIST dataset, the correlated data sources are from images separated by class labels. Each data source only contains the images of the same digit. First, we compare our result, labeled as *Distributed*, to the case where all data are trained with one encoder and one decoder jointly, labeled as *Joint*. The *Joint* curve is approximated as the theoretical upper bound of performance. Second, we compare our result to the case where each data source is trained with a separate pair of encoder and decoder, labeled as *Separate*. In Fig. 4a and 4b, we illustrate the Pearson’s correlation matrix among MNIST images split by random subsets and labels. It shows that the pixels of MNIST images are moderately correlated. Inspired by DSC, it is therefore possible to take advantage of their dependencies by training distributed encoders and a joint decoder. Our experimental studies in the following sections consist of three aspects. We first experiment (2, 4, 8, 10) number of distributed data sources with different correlations. We then show the robustness of our distributed framework in the absence of a number of distributed sources.

To address the advantage of our DNN-based DSC framework, we experiment distributed sources with different correlations. The distributed encoders are labeled as $1, 2, \dots, m$. For example, when $m = 2$, we only use first two subsets of images of digit 0 and 1. We show the result of data sources distributed by random subsets in Fig. 5 and by class labels in Fig. 6. The curves of distributed encoders show that the performance of training distributed encoders and joint decoder can be very close to the theoretical limit. As the number of encoders grows, the performance decreases a little, but still dominantly outperforms training codecs for each data source separately. Results of images split by random subsets also outperform images split by class labels, it may relate to the constant correlation as shown in 4a. The results show that our Deep DSC framework can benefit from dependencies among an arbitrary number of data sources. Our data-driven DSC framework, unlike classical DSC code design, once deployed, does not require synchronization of data sources. In classical DSC code design, if syndrome bits $H(X|Y)$ are used and the data source Y is accidentally blocked, we will not be able to decode the data source X . In our data-driven framework,

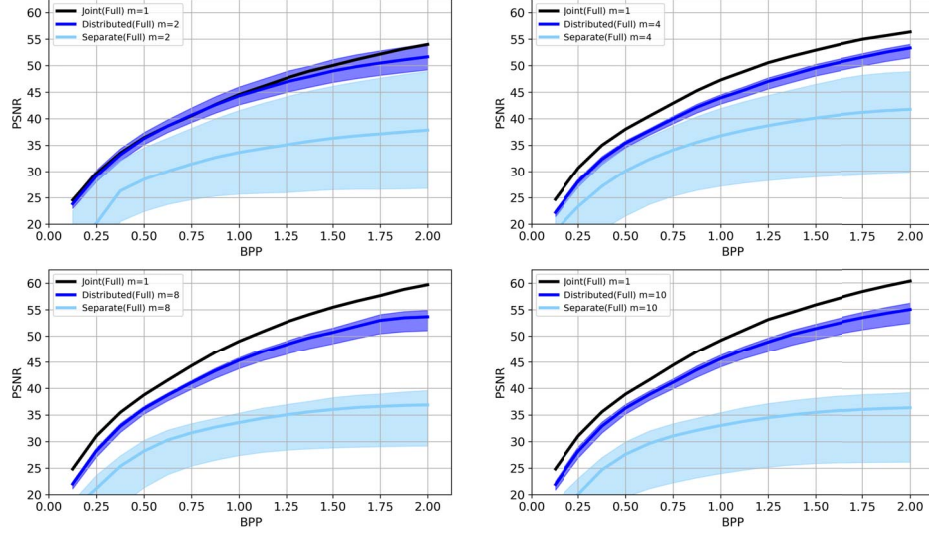


Figure 6: Rate-distortion curves for data sources distributed by class labels with $T = 16$ for all sources.

even only one of the distributed encoders is functional, it can still benefit from its dependencies with other sources because their dependencies are already trained by the model parameters. All our experiments show that distributed encoders not only dominate separately trained codecs but also have narrower confidence bands. As the number of encoders increases, the confidence bands of separately trained codecs become wider because each separate codec can only access very limited amount of data and thus suffer from overfitting.

5 Conclusion

We introduced a data-driven Distributed Source Coding framework based on Distributed Recurrent Autoencoder for Scalable Image Compression (DRASIC). Compared with classical code designs, our method has the following advantages. First, instead of explicitly estimating the correlations among data sources before designing codes, the proposed data-driven approach can *simultaneously learn the dependencies and compress*. Given enough training data, our method can handle an arbitrary number of sources with arbitrary correlations. Second, we showed the robustness of our framework. Unlike classical code designs which often require sophisticated data source synchronization, each distributed encoder of our model, once trained and deployed, can be used *independently and asynchronously* of others. Each data source equipped with less data, fewer number of iterations, and smaller computational power can still approach the theoretical limit of compression obtained by pulling all the data. Last but not least, our recurrent model can reconstruct images efficiently even at low compression quality.

We point out two interesting directions of future work. First, the compression quality of the proposed architecture can be further improved by introducing spatially adaptive weights over different iterations, e.g. by using context models for adaptive arithmetic coding. Second, the network architecture can be further extended to handle time-dependent data sources.

References

- [1] George Toderici, Sean M O'Malley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell, and Rahul Sukthankar, "Variable rate image compression with recurrent neural networks," *arXiv preprint arXiv:1511.06085*, 2015.
- [2] Johannes Ballé, Valero Laparra, and Eero P Simoncelli, "End-to-end optimized image compression," *arXiv preprint arXiv:1611.01704*, 2016.
- [3] Karol Gregor, Frederic Besse, Danilo Jimenez Rezende, Ivo Danihelka, and Daan Wierstra, "Towards conceptual compression," in *Advances In Neural Information Processing Systems*, 2016, pp. 3549–3557.
- [4] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell, "Full resolution image compression with recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5306–5314.
- [5] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár, "Lossy image compression with compressive autoencoders," *arXiv preprint arXiv:1703.00395*, 2017.
- [6] Nick Johnston, Damien Vincent, David Minnen, Michele Covell, Saurabh Singh, Troy Chinen, Sung Jin Hwang, Joel Shor, and George Toderici, "Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks," *structure*, vol. 10, pp. 23, 2017.
- [7] Dong Liu, Haichuan Ma, Zhiwei Xiong, and Feng Wu, "CNN-based DCT-like transform for image compression," in *International Conference on Multimedia Modeling*. Springer, 2018, pp. 61–72.
- [8] Mu Li, Wangmeng Zuo, Shuhang Gu, Debin Zhao, and David Zhang, "Learning convolutional networks for content-weighted image compression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3214–3223.
- [9] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool, "Conditional probability models for deep image compression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4394–4402.
- [10] Zixiang Xiong, Angelos D Liveris, and Samuel Cheng, "Distributed source coding for sensor networks," *IEEE signal processing magazine*, vol. 21, no. 5, pp. 80–94, 2004.
- [11] Bernd Girod, Anne Margot Aaron, Shantanu Rane, and David Rebollo-Monedero, "Distributed video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 71–83, 2005.
- [12] Milica Stojanovic and James Preisig, "Underwater acoustic communication channels: Propagation models and statistical characterization," *IEEE communications magazine*, vol. 47, no. 1, pp. 84–89, 2009.
- [13] Raimondo Schettini and Silvia Corchs, "Underwater image processing: state of the art of restoration and image enhancement methods," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, pp. 14, 2010.
- [14] Prakash Ishwar, Rohit Puri, Kannan Ramchandran, and S Sandeep Pradhan, "On rate-constrained distributed estimation in unreliable sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 765–775, 2005.
- [15] Jin-Jun Xiao, Alejandro Ribeiro, Zhi-Quan Luo, and Georgios B Giannakis, "Distributed compression-estimation using wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 27–41, 2006.
- [16] Christine Guillemot, Fernando Pereira, Luis Torres, Touradj Ebrahimi, Riccardo Leonardi, and Joern Ostermann, "Distributed monoview and multiview video coding," *IEEE Signal Processing Magazine*, vol. 24, no. 5, pp. 67–76, 2007.
- [17] Nicolas Gehrig, *Distributed source coding of multi-view images*, Ph.D. thesis, University of London, 2008.

- [18] David Slepian and Jack Wolf, “Noiseless coding of correlated information sources,” *IEEE Transactions on information Theory*, vol. 19, no. 4, pp. 471–480, 1973.
- [19] T Cover, “A proof of the data compression theorem of slepian and wolf for ergodic sources (corresp.),” *IEEE Transactions on Information Theory*, vol. 21, no. 2, pp. 226–228, 1975.
- [20] Aaron Wyner and Jacob Ziv, “The rate-distortion function for source coding with side information at the decoder,” *IEEE Transactions on information Theory*, vol. 22, no. 1, pp. 1–10, 1976.
- [21] S Sandeep Pradhan and Kannan Ramchandran, “Distributed source coding using syndromes (DISCUS): Design and construction,” *IEEE transactions on information theory*, vol. 49, no. 3, pp. 626–643, 2003.
- [22] Gregory K Wallace, “The JPEG still picture compression standard,” *IEEE transactions on consumer electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.
- [23] Athanassios Skodras, Charilaos Christopoulos, and Touradj Ebrahimi, “The JPEG 2000 still image compression standard,” *IEEE Signal processing magazine*, vol. 18, no. 5, pp. 36–58, 2001.
- [24] Google, “WebP: Compression techniques,” 2010.
- [25] Fabrice Bellard, “BPG image format,” 2014.
- [26] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al., “Conditional image generation with pixelcnn decoders,” in *Advances in neural information processing systems*, 2016, pp. 4790–4798.
- [27] Oren Rippel and Lubomir Bourdev, “Real-time adaptive image compression,” in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 2922–2930.
- [28] Ken M Nakanishi, Shin-ichi Maeda, Takeru Miyato, and Daisuke Okanohara, “Neural multi-scale image compression,” in *Asian Conference on Computer Vision*. Springer, 2018, pp. 718–732.
- [29] Cagatay Dikici, Radhouane Guermazi, Khalid Idrissi, and Atilla Baskurt, “Distributed source coding of still images,” in *Signal Processing Conference, 2005 13th European*. IEEE, 2005, pp. 1–5.
- [30] Rohit Puri and Kannan Ramchandran, “PRISM: A new robust video coding architecture based on distributed compression principles,” in *Proceedings of the annual allerton conference on communication control and computing*. Citeseer, 2002.
- [31] Anne Aaron, Rui Zhang, and Bernd Girod, “Wyner-Ziv coding of motion video,” in *Signals, Systems and Computers, 2002. Conference Record of the Thirty-Sixth Asilomar Conference on*. IEEE, 2002, vol. 1, pp. 240–244.
- [32] Qian Xu and Zixiang Xiong, “Layered Wyner-Ziv video coding,” *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3791–3803, 2006.
- [33] D Schonberg, Kannan Ramchandran, and S Sandeep Pradhan, “Distributed code constructions for the entire slepian-wolf rate region for arbitrarily correlated sources,” in *Data Compression Conference, 2004. Proceedings. DCC 2004*. IEEE, 2004, pp. 292–301.
- [34] Robert Franzén, “Kodak lossless true color image suite: Photocd pcd0992,” 2002.
- [35] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al., “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [36] Diederik P Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.