**What were the major goals and objectives of the project?**

**What was accomplished towards achieving these goals?**

We have two major activities for this reporting period. On the one hand, we focus on establish a systematic approach for general and practical distributed code design, especially in the presence of an arbitrary number of nontrivial data sources with arbitrary correlation. Specifically, in the context of Distributed Source Coding, we use Deep Learning approach to design data-driven image compression models. Our method performs very close to the theoretical limit where we encode all data sources together. On the other hand, we naturally extend neural network-based image compression model to handle image classification problems based on compressed data. Image classification are usually performed on decoded images instead of coded space. However, our result shows that with it is possible to jointly compress and classify images. The decoder can not only reconstruct images but also efficiently classify from the coded space with a two-layer LSTM network.

*Distributed Lossy Image Compression with Recurrent Networks*

We propose a new architecture for distributed image compression from a group of distributed data sources. The proposed architecture, which we refer to as symmetric Encoder-Decoder Convolutional Recurrent Neural Network, is able to significantly outperform the state-of-the-art compression techniques such as JPEG on rate-distortion curves. We also show that by training distributed encoders and joint decoders on correlated data sources, the performance of compression is much better than that by training codecs separately. For 10 distributed sources, our distributed system remarkably performs within 2 dB peak signal-to-noise ratio (PSNR) of that of a single codec trained with all data sources. We experiment distributed sources with different correlations and show how our methodology well matches the Slepian-Wolf Theorem in Distributed Source Coding (DSC) [1]. Our method is also shown to be robust to the lack of presence of encoded data from a number of distributed sources. To our best knowledge, this is the first data-driven DSC framework for general distributed code design with Deep Learning.

Our compression network consists of an encoder, a binarizer, and a decoder. The activation function following each Convolutional Neural Network (CNN) module is tanh (illustrated in Fig. 1 and 2). 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 transformed 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.

We use MNIST dataset consisting of 60,000 training and 10,000 testing grayscale handwritten digit images. We evaluate all models with the metric Peak Signal to Noise Ratio (PSNR) against the Bit Per Pixel (BPP). We compare the compression quality of our model to the baseline model [2] and classical codecs like JPEG [3], JPEG2000 [4] and BPG [5] in Fig. 3. We show the result of data sources distributed by random subsets in Fig. 4, 6 and by class labels in Fig. 5, 7. In the first case, each distributed encoder is trained with a subset of data and only the decoder can access the codes compressed from each data source. In the second case, we distribute data sources by class labels. For example, the first encoder can only access digit data of label zero. We run experiments with (2, 4, 8, 10) number of distributed sources. 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. We test each distributed encoder separately with all test data. The solid curves are the average performance across all encoders. At each iteration, the confidence band is determined by the best and worst performance of all encoders.

The advantages of our proposed architecture are mainly fourfold. First, the architecture is naturally scalable in the sense that codes can be decoded at more than one compression quality levels, and it allows efficient coding of correlated sources which are

not physically co-located. This is especially attractive in video streaming applications [6, 7]. Second, unlike classical DSC which requires customized code design for different scenarios [8], data-driven DSC framework can handle nontrivial distribution of image sources with arbitrary correlations. Third, the computation complexity of the encoder can be transferred to the decoder, a system of low complexity encoders can be used in a variety of application domains, such as multi-view video coding [9], sensor networks [8], and under-water image processing where communication bandwidth and computational power are quite restricted [9, 10]. Fourth, the distributed framework can be more robust against heterogeneous noises or malfunctions of encoders, and such robustness can be crucial in, e.g., unreliable sensor networks [9, 12, 13].

*Classification Based on Compressed Data*

In a band-limited scenario, highly compressed codes are transmitted and used for different tasks. For detecting changes, modeling, representation, etc. in distributed setting, it may not be possible/energy efficient to transmit the entire data to a processing center. Also, it may not be energy efficient to decompress the data and process it for signals of interest. Decentralized perception can be more efficient if a coarse processing of compressed data is performed first. Up on suspecting a signal of interest, full resolution data can be either communicated or full decompression can be performed.

We naturally extend our model to classify images as we can train our decoder and classifier jointly as illustrated in Fig. 8. The classifier is a two-layer LSTM network which takes codes as input and produces class labels. We evaluate our result with MNIST and SVHN dataset. We show the ROC curves of our classifier against different compression qualities in Fig. 9 and Fig. 10. As we gather more codes, the classification accuracy also grows as expected. However, the result shows that we only need 0.25 BPP for MNIST dataset to achieve 98% classification accuracy. Compared to the state-of-the-art Deep Learning classifier which usually takes a high dimensional feature vector, our result shows that it is possible to produce coarse or even accurate predictions with very little information bits while still be able to reconstruct original signals.

**What opportunities for training and professional development did the project provide?**

**How were the results disseminated to communities of interest?**

**What do you plan to do during the next reporting period to accomplish the goals and objectives?**

There are multiple extensions we can make during the next reporting period.

First, we can research further on image compression with deep learning. Since this area is relatively new, we believe there still exists room of improvement. For example, instead of relying on multiple residual iterations of recurrent models, some works concentrate on using L1 penalty to sparsify codes and the entropy compression procedure can be jointly optimized with PixelCNN [14]. Second, we can also further investigate better Deep Distributed Source Coding framework. To enforce low complexity encoding and high complexity decoding, we can use distributed encoders with a smaller number of parameters, while we can maintain free parameters for each distributed code sources in the joint decoder. Third, we can further develop our second activity from classification on compressed data to clustering on compressed data. Our compressive autoencoder can be combined with clustering method in Deep Learning and thus perform clustering or semi-supervised classification on compressed data. We can also combine our image compression method with more computer vision tasks like image object localization, segmentation and caption. It is possible to provide a trade-off between task specific performance and the amount of available information. Finally, we can combine our two activities. What we propose for multi-task learning on compressed data can also be applied in a distributed manner. Therefore, each pair of distributed encoder and decoder can maintain different trade-off between the task performance and the amount of acquired information.

**Honors: What honors or awards were received under this project in this reporting period**

**References**

[1] Slepian, David, and Jack Wolf. "Noiseless coding of correlated information sources." IEEE Transactions on information Theory 19.4 (1973): 471-480.

[2] G. Toderici, D. Vincent, N. Johnston, S. Jin Hwang, D. Minnen, J. Shor, and M. 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. 1, 4, 5

[3] G. K. Wallace, “The JPEG still picture compression standard,” IEEE transactions on consumer electronics, vol. 38, no. 1, pp. xviii–xxxiv, 1992. 2, 5

[4] A. Skodras, C. Christopoulos, and T. Ebrahimi, “The JPEG 2000 still image compression standard,” IEEE Signal processing magazine, vol. 18, no. 5, pp. 36–58, 2001. 2, 5

[5] F. Bellard, “BPG image format,” 2014. [Online]. Available: https://bellard.org/bpg/ 2, 5

[6] C. Guillemot, F. Pereira, L. Torres, T. Ebrahimi, R. Leonardi, and J. Ostermann, “Distributed monoview and multiview video coding,” IEEE Signal Processing Magazine, vol. 24, no. 5, pp. 67–76, 2007. 2

[7] N. Gehrig, “Distributed source coding of multi-view images,” Ph.D. dissertation, University of London, 2008. 2

[8] Z. Xiong, A. D. Liveris, and S. Cheng, “Distributed source coding for sensor networks,” IEEE signal processing magazine, vol. 21, no. 5, pp. 80–94, 2004. 1, 2

[9] B. Girod, A. M. Aaron, S. Rane, and D. RebolloMonedero, “Distributed video coding,” Proceedings of the IEEE, vol. 93, no. 1, pp. 71–83, 2005. 1, 2

[10] M. Stojanovic and J. Preisig, “Underwater acoustic communication channels: Propagation models and statistical characterization,” IEEE communications magazine, vol. 47, no. 1, pp. 84–89, 2009. 2

[11] R. Schettini and S. Corchs, “Underwater image processing: state of the art of restoration and image enhancement methods,” EURASIP Journal on Advances in Signal Processing, vol. 2010, p. 14, 2010. 2

[12] P. Ishwar, R. Puri, K. Ramchandran, and S. S. 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. 2

[13] J.-J. Xiao, A. Ribeiro, Z.-Q. Luo, and G. B. Giannakis, “Distributed compression-estimation using wireless sensor networks,” IEEE Signal Processing Magazine, vol. 23, no. 4, pp. 27–41, 2006. 2

[14] Van den Oord, Aaron, et al. "Conditional image generation with pixelcnn decoders." Advances in neural information processing systems. 2016.