DEEP JOINT SOURCE-CHANNEL CODING FOR WIRELESS IMAGE TRANSMISSION

Eirina Bourtsoulatze

Electronic and Electrical Engineering
University College London, London, UK

David Burth Kurka and Deniz Gündüz

Electrical and Electronic Engineering
Imperial College London, London, UK

ABSTRACT

We propose a novel joint source and channel coding (JSCC) scheme for wireless image transmission that departs from the conventional use of explicit source and channel codes for compression and error correction, and directly maps the image pixel values to the complex-valued channel input signal. Our encoder-decoder pair form an *autoencoder* with a non-trainable layer in the middle, which represents the noisy communication channel. Our results show that the proposed deep JSCC scheme outperforms separation-based digital transmission at low signal-to-noise ratio (SNR) and low channel bandwidth regimes in the presence of additive white Gaussian noise (AWGN). More strikingly, deep JSCC does not suffer from the "cliff effect" as the channel SNR varies with respect to the SNR value assumed during training. In the case of a slow Rayleigh fading channel, deep JSCC can learn to communicate without explicit pilot signals or channel estimation, and significantly outperforms separation-based digital communication at all SNR and channel bandwidth values.

Index Terms— Deep neural networks, joint source-channel coding, wireless image transmission.

1. INTRODUCTION

Conventional wireless image communication systems employ a two-step encoding process, which combines a source code to remove redundant information, and an error correcting code to protect the transmitted information against errors introduced by the wireless channel. According to the Shannon's *separation theorem* [1], this separate architecture is theoretically optimal in the asymptotic limit of infinitely long code blocks. In practice, highly efficient compression algorithms and capacity achieving channel codes are employed to achieve near-optimal performance for reasonably long blocklengths. However, many emerging applications from the Internet-of-things to autonomous driving and to tactile Internet require transmission of image data under extreme latency, bandwidth and/or energy constraints, which preclude computationally demanding long-blocklength state-of-the-art source and channel coding techniques.

Deep learning (DL) based methods, and, particularly, autoencoders [2, 3], have recently shown remarkable results in image compression, achieving or even surpassing the performance of state-of-the-art lossy compression algorithms [4–6]. The advantage of DL-based methods for lossy compression versus conventional compression algorithms lies in their ability to extract complex features from the training data thanks to their deep architecture, and the fact that their model parameters can be trained efficiently on large datasets through backpropagation. At the same time, similarities between the autoencoder architecture and the digital communication systems have motivated the modelling of end-to-end communication systems using the autoencoder architecture [7, 8]. Some examples of such designs include decoder design for existing channel codes [9, 10], blind channel equalization [11], learning physical layer signal representation for SISO [8] and MIMO [12] systems, OFDM systems [13, 14], and JSCC of text messages [15].

In this work, we leverage the recent success of unsupervised DL methods in image compression and communication system design to propose a novel DL-based JSCC technique for image transmission over wireless communication channels. Our coding algorithm directly maps the image pixel values to the complex-valued channel input symbols. In our end-to-end design, the encoding and decoding functions are parameterized by two convolutional neural networks (CNNs) and the communication channel is incorporated in the neural network (NN) architecture as a non-trainable layer; hence, the name *deep JSCC*. We show through experiments that our solution achieves superior performance in low signal-to-noise ratio (SNR) regimes and for limited channel bandwidth, over a time-invariant additive white Gaussian noise (AWGN) channel. We also demonstrate that our approach is resilient to variations in channel conditions, and does not suffer from abrupt quality degradations, known as the "cliff effect" in digital communication systems. This latter property is particularly attractive when broadcasting the same image to multiple receivers with different channel qualities, or when transmitting to a single receiver over an unknown fading channel. Indeed, we show that the proposed deep JSCC scheme achieves a remarkable performance over a slow Rayleigh fading channel despite the lack of explicit pilot signals or channel state information at either side of the communication system, and outperforms a separation-based digital transmission scheme even at high SNR and large channel bandwidth scenarios.

2. PROBLEM FORMULATION

We consider image transmission over a point-to-point noisy wireless communication channel. The transmitter maps the input image $\boldsymbol{x} \in \mathbb{R}^n$ to a vector of complex-valued channel input symbols $\boldsymbol{z} \in \mathbb{C}^k$, which must satisfy a transmit power

This work has been funded by the European Research Council (ERC) through the Starting Grant BEACON (No. 725731) and by a Marie Skłodowska-Curie fellowship (No. 750254).

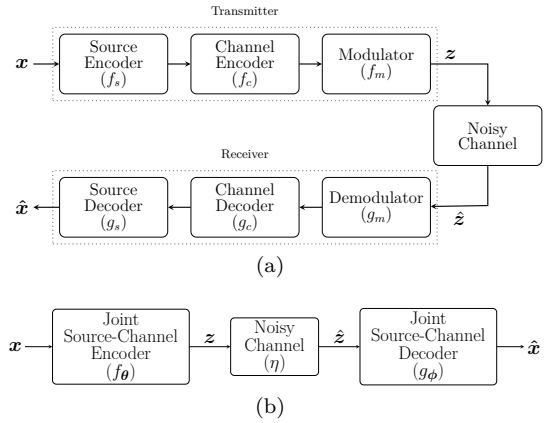

Fig. 1. Block diagram of the point-to-point image transmission system: (a) conventional processing pipeline and (b) proposed deep JSCC algorithm.

constraint. Following the JSCC literature, we will call the image dimension n as the *source bandwidth*, and the channel dimension k as the *channel bandwidth*. We typically have $k < n$, which is called *bandwidth compression*. The channel degrades the signal quality due to noise, fading, interference or other channel impairments. The corrupted output of the communication channel $\hat{\boldsymbol{z}} \in \mathbb{C}^k$ is fed to the receiver, which produces an approximate reconstruction of the original input image, $\hat{\boldsymbol{x}} \in \mathbb{R}^n$.

Fig. 1a shows the encoding and decoding processes of a typical digital transmission system [16]. Although each of the modules in this processing pipeline may be highly optimized, the performance may suffer severely when the channel conditions differ from those for which the system has been optimized. When the channel capacity drops below the designed channel code rate, the error probability goes to 1 (due to strong converse for channel coding), and the receiver cannot receive the correct channel codeword. This leads to a failure in source decoder as well, resulting in a significant reduction in the reconstruction quality. The separate design cannot benefit from improved channel conditions either; that is, once the source and channel coding rates are fixed for a target channel quality, no matter how much the channel improves beyond this target, the reconstruction quality remains the same. These two characteristics are known as the "cliff effect" and various JSCC schemes have been proposed in the literature to overcome this inefficiency [17, 18].

In this paper, we take a radically different approach, and leverage the properties of uncoded transmission [19–21] by directly mapping the real pixel values to complex-valued samples transmitted over the communication channel, as depicted in Fig. 1b. Our goal is to design a JSCC scheme that bypasses the transformation of the pixel values to a sequence of bits, which are then mapped again to real-valued channel inputs; and instead, the proposed scheme directly maps the pixel values to channel inputs. This is similar to recently proposed analog transmission techniques [20, 21] with the main difference of not limiting the mappings to linear transforms.

3. DL-BASED JSCC

Our design is inspired by the recent successful application of deep NNs (DNNs), and autoencoders, in particular, to the problem of image compression [4, 6], as well as by the first promising results in the design of end-to-end communication systems using autoencoder architectures [7, 8].

The block diagram of the proposed JSCC scheme is shown in Fig. 1b. The encoder maps the n-dimensional input image $\boldsymbol{x}$ to a k-length vector of complex-valued channel input samples $\boldsymbol{z}$, which satisfies the average power constraint $\frac{1}{k}\mathbb{E}[\boldsymbol{z}^*\boldsymbol{z}] \leq P$, by means of a deterministic encoding function $f_{\boldsymbol{\theta}} : \mathbb{R}^n \to \mathbb{C}^k$. The encoder function $f_{\boldsymbol{\theta}}$ is parameterized using a CNN with parameters $\boldsymbol{\theta}$; and therefore, can be highly non-linear. The encoder CNN comprises a series of convolutional layers followed by a fully connected (FC) layer and a normalization layer. The convolutional layers extract the image features which are subsequently combined by the FC layer to form the channel input samples. The output $\tilde{\boldsymbol{z}} \in \mathbb{C}^k$ of the FC layer is normalized according to:

$$\boldsymbol{z} = \sqrt{kP}\frac{\tilde{\boldsymbol{z}}}{\sqrt{\tilde{\boldsymbol{z}}^*\tilde{\boldsymbol{z}}}}, \tag{1}$$

where $\tilde{\boldsymbol{z}}^*$ is the conjugate transpose of $\tilde{\boldsymbol{z}}$, such that channel input $\boldsymbol{z}$ satisfies the average transmit power constraint P.

Following the encoding operation, the joint source-channel coded sequence $\boldsymbol{z}$ is sent over the communication channel by directly transmitting the real and imaginary parts of the channel input samples over the I and Q components of the digital signal. The channel introduces random corruption to the transmitted symbols, denoted by $\eta : \mathbb{C}^k \to \mathbb{C}^k$. In order to optimize the communication system in Fig. 1b in an end-to-end manner, the communication channel is incorporated into the overall NN architecture and modelled as a non-trainable layer which is represented by the transfer function $\hat{\boldsymbol{z}} = \eta(\boldsymbol{z}) = \boldsymbol{H}\boldsymbol{z} + \boldsymbol{n}$, where $\boldsymbol{H}$ is the channel gain and $\boldsymbol{n}$ is the additive noise. Other channel models can be incorporated into the end-to-end system in a similar manner with the only requirement that the channel transfer function is differentiable in order to allow gradient computation and error back propagation.

The receiver is a joint source-channel decoder. The decoder maps the corrupted complex-valued signal $\hat{\boldsymbol{z}} = \eta(\boldsymbol{z}) \in \mathbb{C}^k$ directly to an estimate of the original input vector, $\hat{\boldsymbol{x}} \in \mathbb{R}^n$, using a decoding function $g_{\boldsymbol{\phi}} : \mathbb{C}^k \to \mathbb{R}^n$. Similarly to the encoding function, the decoding function is parameterized by the decoder CNN with parameter set $\boldsymbol{\phi}$. The NN decoder inverts the operations performed by the encoder by first passing the noisy received signal $\hat{\boldsymbol{z}}$ through a FC neural layer to obtain the image features. It then applies a series of transpose convolutional layers in order to map the image features to an estimate $\hat{\boldsymbol{x}}$ of the originally transmitted image.

The encoding and decoding functions are designed jointly to minimize the average distortion between the original input image $\boldsymbol{x}$ and its reconstruction $\hat{\boldsymbol{x}}$ produced by the decoder:

$$(\boldsymbol{\theta}^*, \boldsymbol{\phi}^*) = \arg\min_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathbb{E}_{p(\boldsymbol{x},\hat{\boldsymbol{x}})}[d(\boldsymbol{x}, \hat{\boldsymbol{x}})], \tag{2}$$

where $d(\boldsymbol{x}, \hat{\boldsymbol{x}})$ is a given distortion measure, and $p(\boldsymbol{x}, \hat{\boldsymbol{x}})$ is the joint probability distribution of the original and reconstructed images. Since the true distribution of the input data $p(\boldsymbol{x})$ is

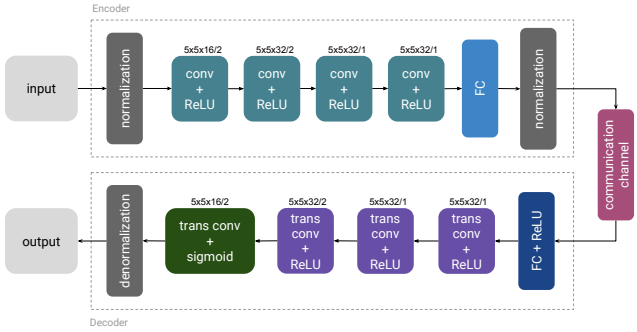

Fig. 2. Encoder and decoder NN architectures used in the implementation of the proposed deep JSCC scheme.

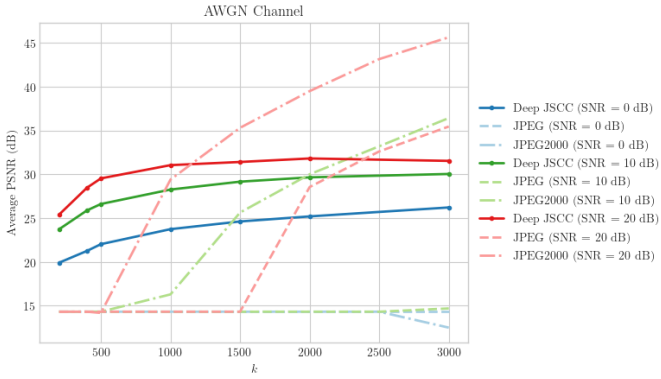

Fig. 3. Performance of the learned JSCC algorithm on test images over an AWGN channel with respect to the channel bandwidth, k, for different SNR values. For each case, the same target SNR value is used in training and evaluation.

often unknown, an analytical form of the expected distortion in Eq. (2) is also unknown. We, therefore, estimate the expected distortion by sampling from an available dataset.

4. EVALUATION

To evaluate our proposed deep JSCC scheme, we use the NN architecture depicted in Fig. 2. At the encoder, the normalization layer pre-processes the input image by dividing it by 255, producing pixel values in the $[0, 1]$ range. The notation $F \times F \times K/S$ denotes a convolutional layer with K filters of spatial extent (or size) F and stride S. The output of the FC layer is of size $2k$. The FC layer is followed by another normalization layer which enforces the average power constraint specified in Eq. (1). The decoder inverts the operations performed by the encoder. The noisy channel output samples are fed into the FC layer of input size $2k$, and then into the transpose convolutional layers, which progressively transform the corrupted image features into an estimate of the original input image, while upsampling it to the correct resolution. The hyperparameters of the decoder layers mirror the corresponding values of the encoder layers (Fig. 2). The denormalization layer multiplies the output values by 255, and rounds them in order to generate pixel values within the $[0, 255]$ range.

The above architecture is implemented in Tensorflow [22]. The training data consists of the $N = 50000$ CIFAR-10 32×32 training images [23] combined with random realizations of the channel under consideration. We use the Adam optimization framework [24], which is a form of stochastic gradient descent, with learning rate 0.0001 and a mini-batch size of 128 samples. Our loss function is the average mean squared error (MSE) between the original input image $\boldsymbol{x}$ and the reconstruction $\hat{\boldsymbol{x}}$ at the output of the decoder, defined as

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} ||\boldsymbol{x} - \hat{\boldsymbol{x}}||^2. \tag{3}$$

We first investigate the performance of our proposed deep JSCC algorithm in the additive white Gaussian noise setting, *i.e.,* the channel transfer function is $\eta(\boldsymbol{z}) = \boldsymbol{z} + \boldsymbol{n}$. Without loss of generality, we assume that the channel input and noise samples are real-valued, *i.e.* $\text{Im}\{\boldsymbol{z}\} = \text{Im}\{\boldsymbol{n}\} = \boldsymbol{0}$ and $\text{Re}\{\boldsymbol{n}\} \sim \mathcal{N}(0, N_0\boldsymbol{I}_k)$. Note that in this case the input/output of the FC layer in the encoder/decoder is of size k. We set the average power constraint to $P = 1$, and vary the channel SNR by varying the noise variance N_0.

The performance of the deep JSCC algorithm is quantified in terms of the PSNR of the reconstructed images at the decoder. We compare our algorithm with a digital transmission scheme, which employs JPEG or JPEG2000 for compression followed by a capacity-achieving channel coding and modulation scheme. According to the Shannon's separation theorem, the maximum number of bits per source sample that can be transmitted reliably is $R = \frac{k}{n}C$, where C is the channel capacity. Let R_{min} be the minimum rate beyond which compression results in complete loss of information and the original image cannot be reconstructed. For all rates $R < R_{min}$, we assume that the image is reconstructed to the mean value of all its pixels. For rates $R \geq R_{min}$, we compress the images at the largest rate R' that satisfies $R' \leq R$.

Fig. 3 illustrates the performance comparison between the proposed deep JSCC algorithm and the digital schemes over an AWGN channel as a function of the channel bandwidth k in different SNR regimes. Here we assume a different encoder-decoder NN pair is trained for each SNR and channel bandwidth value. The threshold behavior of the digital schemes in the figure is due to the fact that the compression rate is below the minimum required number of bits per pixel, R_{min}, to recover a meaningful reconstruction of the images. Indeed, we assume capacity-achieving channel codes for each SNR value despite the short blocklength. We observe that, for limited channel bandwidth ($k \in [200, 1000]$), the performance of deep JSCC is considerably above the one achieved by JPEG and JPEG2000. The performance of the digital separate source and channel coding scheme with JPEG2000 improves significantly if we increase the channel bandwidth to $k = 1000$; however, the proposed scheme still outperforms this reference performance for all but very high SNR values.

We next study the robustness of the proposed deep JSCC scheme to variations in channel conditions. Fig. 4 illustrates the average PSNR of the reconstructed images versus the SNR of the additive white Gaussian noise channel for $k = 500$. Each curve is generated by training our end-

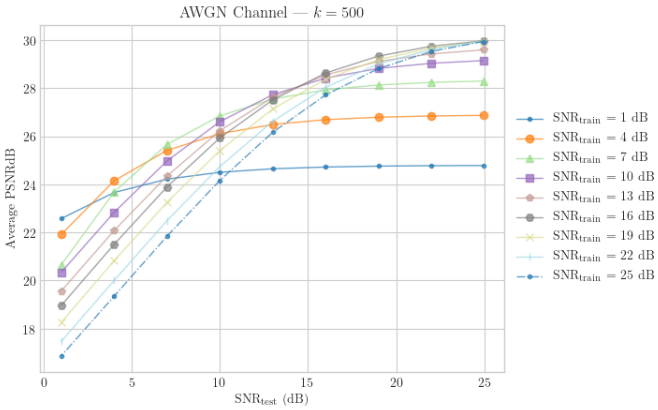

Fig. 4. Performance of the learned JSCC algorithm with respect to channel SNR over an AWGN channel with channel bandwidth $k = 500$.

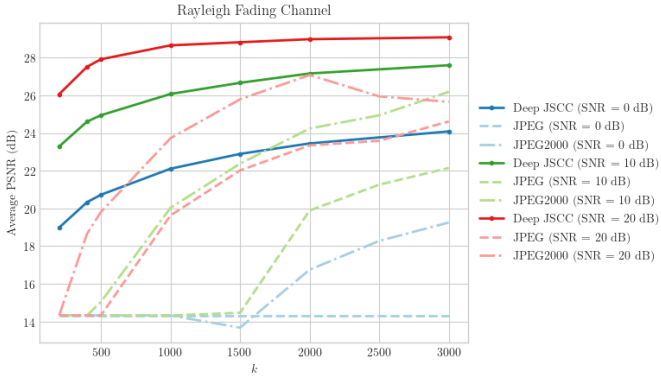

Fig. 5. Average PSNR with respect to channel bandwidth, k, over a slow Rayleigh fading channel for different SNR values. Same target SNR value is used for both training and testing.

to-end system for a specific channel SNR value, denoted as SNR_{train}, and then evaluating the performance of the learned encoder/decoder parameters on the test images for varying SNR values, denoted as SNR_{test}. When the channel conditions are worse than those for which the encoder/decoder have been optimized ($\mathrm{SNR}_{test} < \mathrm{SNR}_{train}$), our deep JSCC algorithm does not suffer from the "cliff effect" observed in digital systems, where the quality of the decoded signal drops sharply when SNR_{test} drops below SNR_{train}. Our deep JSCC scheme is more robust to channel quality fluctuations, and exhibits a gradual performance degradation as the channel deteriorates. Such behavior is akin to the performance of an analog scheme [17, 19, 21], and is attributed to the capability of the autoencoder to map similar images/features to nearby points in the channel input signal space. On the other hand, when SNR_{test} increases above SNR_{train}, we observe a gradual improvement in the average quality of the reconstructed images before the performance finally saturates as SNR_{test} increases beyond a certain value. The performance in the saturation region is driven solely by the amount of compression implicitly decided during the training phase for the target value SNR_{train}. It is worth noting that performance saturation does not occur at $\mathrm{SNR}_{test} = \mathrm{SNR}_{train}$ as in digital image/video transmission systems, for which an improvement beyond the target SNR does not help in terms of the end-to-end PSNR performance, but at $\mathrm{SNR}_{test} > \mathrm{SNR}_{train}$.

Finally, we present the performance of our deep JSCC scheme under the assumption of a slow Rayleigh fading AWGN channel. Specifically, we assume that the channel gain is sampled from a Rayleigh distribution and remains constant for the duration of the transmission of the whole image, and changes independently to another state for the next image. In this case, the channel input samples are complex valued and the channel transfer function is $\eta(\boldsymbol{z}) = \mathrm{diag}(\boldsymbol{h})\boldsymbol{z} + \boldsymbol{n}$, where $\boldsymbol{h} \sim \mathcal{CN}(0, H_c\boldsymbol{I}_k)$ and $\boldsymbol{n} \sim \mathcal{CN}(0, N_0\boldsymbol{I}_k)$. We do not assume channel state information either at the receiver or the transmitter, or consider the transmission of pilot signals. We set $H_c = 1$, $P = 1$, and vary the noise variance N_0 to emulate time-varying channel SNR.

In Fig. 5, we plot the performance of the proposed deep JSCC algorithm over a slow Rayleigh fading channel as a function of the channel bandwidth, k, for different average SNR values. Note that, due to the lack of channel state information, the capacity of this channel in the Shannon sense is zero, since no positive rate can be guaranteed reliable transmission at all channel conditions; that is, for any positive transmission rate, the channel capacity will be below the transmission rate with a non-zero probability. Therefore, for digital transmission, we assume that the transmitter transmits at rate that is equal to the capacity of the complex AWGN channel at the average SNR value. If the channel capacity is below this value, an outage occurs, and the mean pixel values are used for reconstruction, i.e., maximum distortion is reached. If the channel capacity is above the transmission rate, the transmitted codeword can be decoded reliably. This scheme inherently assumes that the receiver knows the channel realization, which is not the case for deep JSCC. We observe that deep JSCC beats the benchmark digital transmission scheme at all SNR and channel bandwidth values. This result emphasizes the benefits of the proposed deep JSCC technique when communicating over a time-varying channel, or multicasting to multiple receivers with different channel conditions.

5. CONCLUSIONS

We have proposed a novel deep JSCC architecture for image transmission over wireless channels. In this architecture, the encoder maps input images directly to channel inputs. The encoder and decoder functions are modeled as complementary DNNs, and trained jointly on the dataset to minimize the average MSE of the reconstructed image. We have shown through extensive numerical simulations that deep JSCC outperforms separation-based schemes, especially for limited channel bandwidth and SNR regimes. More significantly, deep JSCC is shown to provide a graceful degradation of the reconstruction quality with channel SNR. This observation is then used to benefit from the proposed scheme when communicating over a slow fading channel. Despite the absence of pilot signals or explicit channel estimation, deep JSCC performs reasonably well at all average SNR values, and outperforms the proposed separation-based transmission scheme at any channel bandwidth value.

6. REFERENCES

[1] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley-Interscience, 1991.

[2] Y. Bengio, "Learning deep architectures for AI," *Found. and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, Jan. 2009.

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.

[4] L. Theis, W. Shi, A. Cunnigham, and F. Huszár, "Lossy image compression with compressive autoencoders," in *Proc. of the Int. Conf. on Learning Representations (ICLR)*, 2017.

[5] O. Rippel and L. Bourdev, "Real-time adaptive image compression," in *Proc. Int. Conf. on Machine Learning (ICML)*, vol. 70, Aug. 2017, pp. 2922–2930.

[6] J. Balle, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," in *Proc. of Int. Conf. on Learning Representations (ICLR)*, Apr. 2017, pp. 1–27.

[7] T. J. O'Shea, K. Karra, and T. C. Clancy, "Learning to communicate: Channel auto-encoders, domain specific regularizers, and attention," in *Proc. of IEEE Int. Symp. on Signal Processing and Information Technology (ISSPIT)*, Dec. 2016, pp. 223–228.

[8] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, Dec 2017.

[9] H. Kim, Y. Jiang, R. B. Rana, S. Kannan, S. Oh, and P. Viswanath, "Communication algorithms via deep learning," in *Proc. of Int. Conf. on Learning Representations (ICLR)*, 2018.

[10] E. Nachmani, E. Marciano, L. Lugosch, W. J. Gross, D. Burshtein, and Y. Baery, "Deep learning methods for improved decoding of linear codes," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 119–131, Feb 2018.

[11] A. Caciularu and D. Burshtein, "Blind channel equalization using variational autoencoders," in *Proc. IEEE Int. Conf. on Comms. Workshops, Kansas City, MO*, May 2018, pp. 1–6.

[12] T. J. O'Shea, T. Erpek, and T. C. Clancy, "Deep learning based MIMO communications," *arXiv:1707.07980 [cs.IT]*, 2017.

[13] A. Felix, S. Cammerer, S. Dorner, J. Hoydis, and S. ten Brink, "OFDM autoencoder for end-to-end learning of communications systems," in *Proc. IEEE Int. Workshop Signal Proc. Adv. Wireless Commun. (SPAWC)*, Jun. 2018.

[14] H. Ye, G. Y. Li, and B. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, Feb. 2018.

[15] N. Farsad, M. Rao, and A. Goldsmith, "Deep learning for joint source-channel coding of text," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018.

[16] N. Thomos, N. V. Boulgouris, and M. G. Strintzis, "Optimized transmission of JPEG2000 streams over wireless channels," *IEEE Trans. on Image Processing*, vol. 15, no. 1, pp. 54–67, Jan 2006.

[17] D. Gunduz and E. Erkip, "Joint source-channel codes for MIMO block-fading channels," *IEEE Trans. on Information Theory*, vol. 54, no. 1, pp. 116–134, Jan 2008.

[18] I. Kozintsev and K. Ramchandran, "Robust image transmission over energy-constrained time-varying channels using multiresolution joint source-channel coding," *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 1012–1026, April 1998.

[19] T. Goblick, "Theoretical limitations on the transmission of data from analog sources," *IEEE Transactions on Information Theory*, vol. 11, no. 4, pp. 558–567, October 1965.

[20] S. Jakubczak and D. Katabi, "SoftCast: Clean-slate scalable wireless video," in *Proc. of the 48th IEEE Annual Allerton Conf. on Communication, Control, and Computing*, Illinois,USA, Sept. 2010, pp. 530–533.

[21] T. Tung and D. Gunduz, "Sparsecast: Hybrid digital-analog wireless image transmission exploiting frequency domain sparsity," *IEEE Communications Letters*, pp. 1–1, 2018.

[22] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[23] A. Krizhevsky, "Learning multiple layers of features from tiny images," University of Toronto, Tech. Rep., 2009.

[24] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," *arXiv:1412.6980 [cs.LG]*, 2014.