

Successive Refinement of Images with Deep Joint Source-Channel Coding

David Burth Kurka and Deniz Gündüz
Information Processing and Communications Laboratory
Department of Electrical and Electronic Engineering
Imperial College London, London, UK
{d.kurka, d.gunduz}@imperial.ac.uk

Abstract—We introduce deep learning based communication methods for successive refinement of images over wireless channels. We present three different strategies for progressive image transmission with deep JSCC, with different complexity-performance trade-offs, all based on convolutional autoencoders. Numerical results show that deep JSCC not only provides graceful degradation with channel signal-to-noise ratio (SNR) and improved performance in low SNR and low bandwidth regimes compared to state-of-the-art digital communication techniques, but can also successfully learn a layered representation, achieving performance close to a single-layer scheme. These results suggest that natural images encoded with deep JSCC over Gaussian channels are almost successively refinable.

I. INTRODUCTION

We consider progressive transmission of images over a point-to-point wireless channel. In this scenario, an image is transmitted in multiple stages, with each realization improving its quality. Typically, we expect the first layer to be low-quality, but enough to convey the main elements of the content being transmitted. Following layers are then used to enhance the image originally received, by adding more details and components to it [1]. Progressive transmission can be applied to scenarios in which communication is either expensive or urgent. For example, in surveillance applications it may be beneficial to quickly send a low-resolution image to detect a potential threat as soon as possible, while a higher resolution description can be later received for further evaluation or archival purposes. It is also possible that the higher layers can be received by only a subset of the receivers. This may be the case in wireless multicasting of the same image to devices with different resolutions. Progressive transmission would allow low-resolution devices to receive and decode only a limited portion of the channel resources, saving energy, while high-resolution receivers can recover a better quality reconstruction by receiving additional channel resources.

Information theoretically, this problem corresponds to hierarchical joint source-channel coding (JSCC), studied in [2], where the optimality of separation is proven; that is, it is optimal to compress the image into multiple layers using successive refinement source coding [3], where the rate of each layer is dictated by the capacity of the channel

it is transmitted over. In general, successive refinement introduces losses compared to single-layer compression at the highest possible resolution; that is, the adaptation to channel bandwidth comes at a price, although some ideal source distributions are known to be *successively refinable* under certain performance measures, which means that they can be progressively compressed at no rate loss, e.g., Gaussian sources over Gaussian channels. On the other hand, it is known that in practical scenarios JSCC can provide gains compared to separate source and channel code design.

Here, following our previous work [4], we use deep learning (DL) methods, in particular, the autoencoder architecture [5], for the design of an end-to-end progressive image transmission system. In [4], we introduced a novel end-to-end DL-based JSCC scheme for image transmission over wireless communication channels, called the *deep JSCC*, where encoding and decoding functions are parameterized by convolutional neural networks (CNNs) and the communication channel is incorporated into the neural network (NN) architecture as a non-trainable layer. This method achieves remarkable performance in low signal-to-noise ratio (SNR) and limited channel bandwidth, also showing resilience to mismatch between training and test channel conditions and channel variations similarly to analog communications.

DL-based methods are receiving significant attention for the design of novel and efficient coding and modulation techniques. In particular, the similarities between the autoencoder architecture and the digital communication systems have motivated many studies including decoder design for existing channel codes [6], [7], blind channel equalization [8], learning physical layer signal representation for SISO [9] and MIMO [10] systems, OFDM systems [11], [12], JSCC of text messages [13], and JSCC for analog storage [14]. Similar methods have also recently shown notable results in image compression [15]–[17].

We propose three different architectures for progressive deep JSCC with different complexities. The results are remarkable in the sense that progressive transmission in multiple layers introduces a limited performance loss (in terms of average PSNR) compared to single-layer transmission; that is, deep JSCC allows adding new layers with

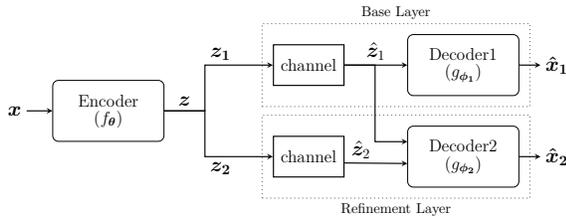


Fig. 1. Progressive wireless image transmission with two layers.

almost no penalties on the performance of the existing layers. This result suggests that natural images transmitted with deep JSCC are successively refinable over Gaussian channels under the PSNR performance measure. This also suggests that deep JSCC not only provides natural adaptation to the channel quality [4], but also to the bandwidth. It is shown in [4] that deep JSCC has better or comparable performance to separate source and channel coding (JPEG2000 followed by high performance channel codes) in single-layer transmission. Here we show that the advantages of deep JSCC extend to progressive transmissions as well.

II. BACKGROUND AND PROBLEM FORMULATION

We consider progressive wireless transmission of images, where the input image $\mathbf{x} \in \mathbb{R}^n$ is transmitted in L layers. Let $\mathbf{z}_i, \hat{\mathbf{z}}_i \in \mathbb{C}^k$ denote the complex channel input and output vectors for the i th layer, $i \in [L] \triangleq [1, \dots, L]$. The receiver outputs a different image reconstruction $\hat{\mathbf{x}}_i$ after receiving the i th layer (using the first i layers). Equivalently, we can consider L virtual receivers corresponding to each layer. See Figure 1 for an illustration of the system model for $L = 2$. We will call the image dimension n as the *source bandwidth*, and the channel dimension k_i as the *bandwidth of channel i* . We will refer to the ratio k_i/n as *bandwidth compression ratio* for the i th layer. An average power constraint is imposed on the transmitted signal at each layer \mathbf{z}_i , $\frac{1}{k_i} \mathbb{E}[\mathbf{z}_i^* \mathbf{z}_i] \leq P$.

The reconstruction after receiving the first i layers is denoted by $\hat{\mathbf{x}}_i \in \mathbb{R}^n$. Its performance is evaluated by the peak signal to noise ratio (PSNR _{i}), which is the inverse of the mean square error (MSE), defined as:

$$\text{MSE}_i = \frac{1}{N} \sum_{i=1}^N \|\mathbf{x} - \hat{\mathbf{x}}_i\|^2; \text{PSNR}_i = 10 \log_{10} \frac{\text{MAX}^2}{\text{MSE}_i},$$

where MAX is the maximum value a pixel can take, which is 255 in our case. Two channel models, the additive white Gaussian noise (AWGN) channel and the slow Rayleigh fading channel, are considered in this work.

The experimental sections explore different schemes for encoding and decoding strategies, but they all share the same CNN architecture for the encoder and decoder components, shown in Figure 2. Once the model is created, the encoder(s) and decoder(s) are trained jointly as unsupervised learning, while the channel is incorporated

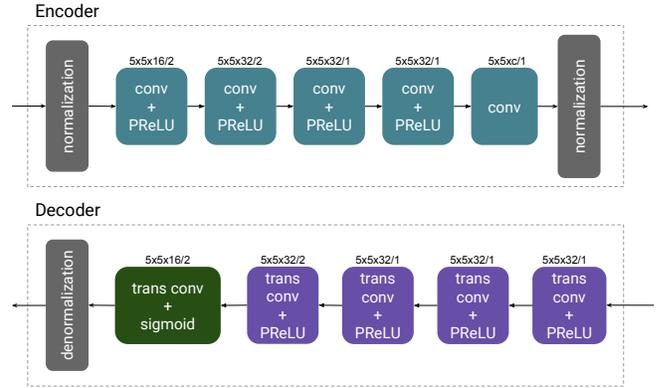


Fig. 2. The encoder and decoder components used in this paper.

to the model as a non-trainable layer, producing random values at every realization. We implement our models in TensorFlow and use the Cifar10 dataset in all the experiments. Previous work [4] demonstrated the efficiency of this architecture for JSCC, creating encoders capable of directly mapping pixels to channel inputs, and decoders that retrieve the underlying image directly from the noisy channel outputs, achieving better or comparable performance compared to the state-of-the-art digital separation-based transmission schemes. In the next sections, we will present different strategies for progressive JSCC of images over wireless channel, and present numerical results.

III. MULTIPLE DECODERS

In the first model we consider a single encoder NN generating at once the complete channel input vector $\mathbf{z} = [\mathbf{z}_1 \dots \mathbf{z}_L]$, which is transmitted in L stages. Then L independent decoder NNs are considered, where the i th decoder uses $\sum_{j=1}^i k_j$ channel output symbols to output a distinct reconstruction of the input image.

The system is modelled as an autoencoder with one encoder and L decoder NNs, with the loss defined as:

$$\mathcal{L} = \frac{1}{L} \frac{1}{N} \sum_{i=1}^L \sum_{j=1}^N d(\mathbf{x}_j, \hat{\mathbf{x}}_{i,j}), \quad (1)$$

where $d(\mathbf{x}, \hat{\mathbf{x}})$ is the MSE distortion.

A. Two-layer model

We first focus on the scenario with $L = 2$ layers, thus requiring the training of only one encoder and two decoders. The second decoder receives the output of both transmissions; and hence, should achieve a better performance.

1) *AWGN Channel*: First, we consider an AWGN channel, with $k_1/n = 1/12$ and $k_2/n = 1/12$. Our experiments consider different channel qualities (specified by the SNR). Thus, a set of encoder and decoders are trained and optimized for each target channel SNR. The results are shown in Figure 3a. Each colour represents the same model trained for a specific SNR, with one curve corresponding

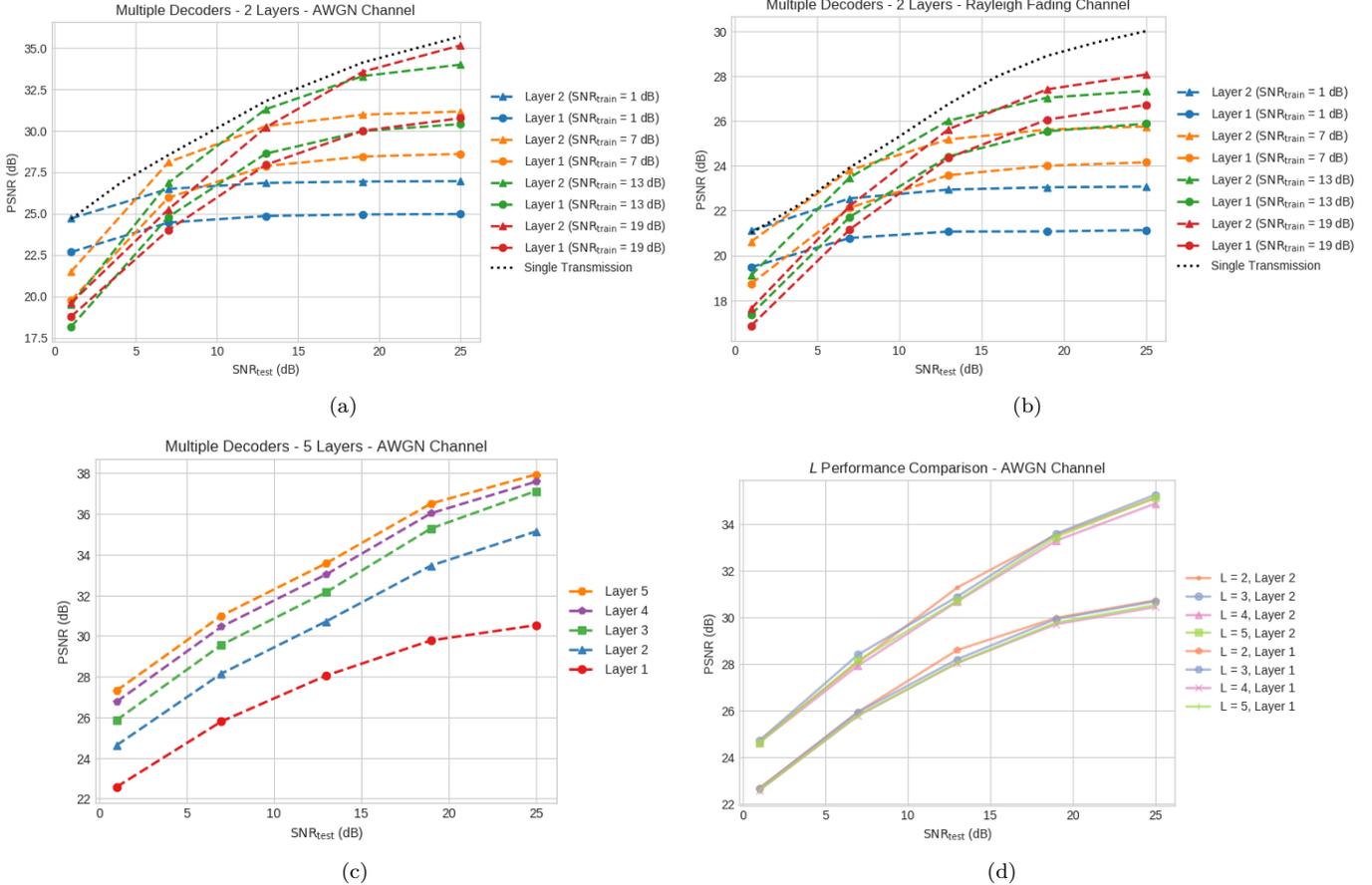


Fig. 3. Performance of multiple decoders scheme on CIFAR-10 test images, with respect to the channel SNR for (a) AWGN channel, $k_1/n = k_2/n = 1/12$; (b) fading channel, $k_1/n = k_2/n = 1/12$; (c) five decoders, AWGN channel, with $k_i/n = 1/12$ for $i \in \{1 \dots 5\}$; (d) performance for two first layers of models trained with different values of L .

to the decoder receiving only the base layer, while the other the decoder receiving both layers. Although the model is optimized for a specific SNR, our results show, for each trained model, evaluations in a range of test SNRs (1-25dB). We see that in all cases, average PSNR₂ is consistently higher than PSNR₁ by 2 to 3 dB, showing that the successive refinement has been achieved.

As a baseline, we compare our results to a single layer transmission scheme, with the same channel bandwidth as the sum of the individual layers, that is $k = k_1 + k_2$. We see that the progressive JSCC scheme can approach the same performance as the single layer, showing that there are no significant losses in the transmission efficiency when the model is adapted for successive refinement schemes.

The evaluation in multiple SNRs, including lower SNRs than the trained SNR, shows that the scheme is robust against channel deterioration, not suffering from the *cliff effect*, but instead presenting graceful degradation. This analog property of the model was already observed in the single layer case in [4]. This behaviour is valid for all other results presented in this paper. However, due to a space limitation, those results will not be explicitly shown.

2) *Fading channel:* We move on to consider the same model on a slow Rayleigh fading channel. Figure 3b shows results for an architecture similar to the one used in Figure 3a. We see that, although the PSNRs are lower than those in the AWGN case due to channel variations, the overall properties of graceful degradation and analog behaviour are still present. Besides, the performance of the deep-JSCC scheme is significantly superior than the state-of-art separation based schemes in the case of fading channels, despite the lack of explicit pilots or channel estimation [4].

Although all the models exhibit similar behaviour over fading channels, we will limit our attention to the AWGN channel in the rest of the paper due to space limitation.

B. Multiple layers

Next, we extend the model to multiple layers. Figure 3c shows the results for five layers, each with bandwidth compression equal to 1/12. For each test SNR, only the highest PSNR obtained is plotted (i.e., the convex hull of the previous plots).

The results show that the addition of new layers increases the overall quality of the transmitted image at

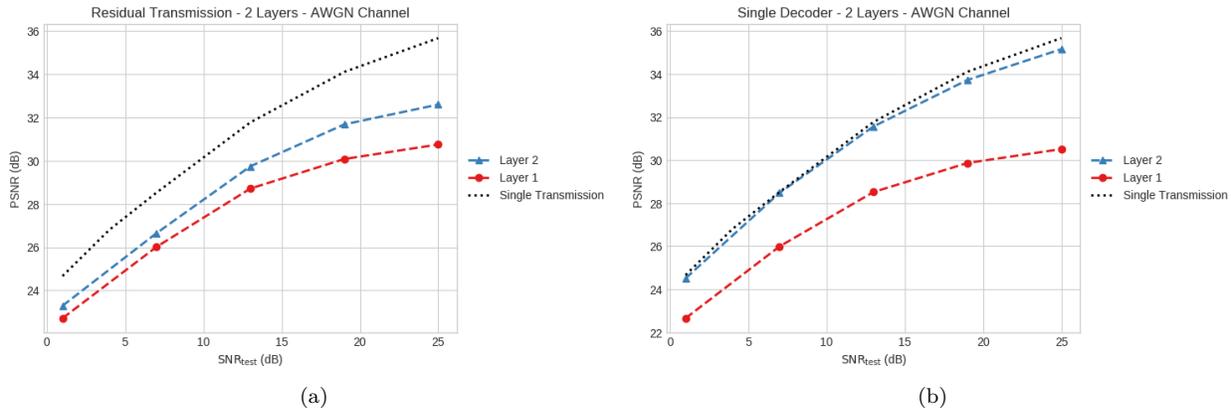


Fig. 4. Performance of different schemes on CIFAR-10 test images, with respect to the channel SNR. (a) Residual transmission scheme, AWGN channel, $k_1/n = k_2/n = 1/12$; (b) Single decoder scheme, AWGN channel, $k_1/n = k_2/n = 1/12$.

every step; although the amount of improvement is diminishing, as the model is able to transmit the main image features with the lower layers, leaving only marginal contributions to the additional layers.

We notice that the introduction of additional layers in the training model has very low impact on the performance of the first layers, if compared to models with smaller values of L . This can be seen in Figure 3d, which compares the performance of the first layer for models trained with $L \in \{2, 3, 4, 5\}$, showing that the loss of adding new layers is negligible. This is rather surprising, given that the code of the first layer is shared by all the layers and is optimized for all layers, as in Eq. 1. The results, therefore, suggest that there is performance independence between layers, justifying the use of as many layers as desired, as long as there are available resources.

IV. RESIDUAL TRANSMISSION

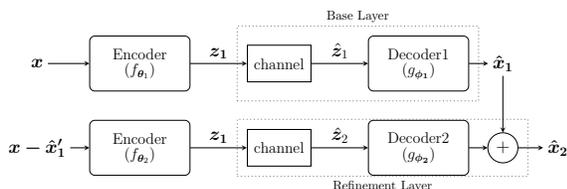


Fig. 5. Residual transmission scheme with two layers.

We proceed our investigation by proposing an alternative scheme. Now, as seen in Figure 5, each transmission is performed by an independent encoder/decoder pair that act in sequence. The first pair (the base layer) is designed to transmit the original image x , retrieving \hat{x}_1 . Then, each subsequent layer computes an estimate of the image reconstructed at the receiver side using all the previous layers, so it can transmit only a residual image:

$$\mathbf{x}_i^{res} = \mathbf{x} - \sum_{j=1}^{i-1} \hat{\mathbf{x}}_j'.$$

We assume that the estimated output $\hat{\mathbf{x}}_j'$ is equal to the actual receiver output during the training phase (i.e., $\hat{\mathbf{x}}_j' = \hat{\mathbf{x}}_j$); however, during evaluation, we consider the receiver is deployed and inaccessible, so the estimation is obtained by averaging independent realizations of the channel and decoder models (i.e., $\hat{\mathbf{x}}_i' = \frac{1}{m} \sum_{i=1}^m \hat{\mathbf{x}}$, where m is the number of independent channel realizations used to estimate receiver's output).

Each encoder/decoder pair is optimized separately, using the result of the previous layers. Although this solution is more computationally expensive, this allows design flexibility, as new layers can be added to the model as they are required, without the need of any change on the previous trained parts. Figure 4a considers a similar scenario as Figure 3a, with AWGN channel and $k_1/n = k_2/n = 1/12$. As expected, the performance of this scheme is significantly worse than the previous one.

A. Feedback channel

Note that the residual encoder does not know the actual realization of the channel of the first transmission, so it has to estimate the residual based on the channel model. The better the estimate of the decoded image of the previous layer is, the better is the quality of the residual transmission. We have estimated the residual image by emulating the channel m times, and encoding the average of the residuals. In Table I we present the model's performance for different m values, and observe that the estimation accuracy increases with m .

The last column in Table I corresponds to the performance when the encoder has perfect channel output feedback; and hence, it can perfectly reconstruct the residual. We see that the performance with perfect channel output feedback is close to the one with two decoders trained jointly. This is in line with the information theoretical results stating that feedback, in general, does not improve the end to end average reconstruction quality in this setting, but it can allow simpler more flexible schemes to be implemented.

TABLE I
PERFORMANCE OF RESIDUAL TRANSMISSION STRATEGY FOR
DIFFERENT CHANNEL ESTIMATION COUNTS m , FOR
 $\text{SNR}_{\text{train}} = \text{SNR}_{\text{test}} = 13\text{dB}$

Realizations	1	10	100	1000	direct feedback
PSNR	29.04	29.68	29.73	29.75	30.64

V. SINGLE DECODER

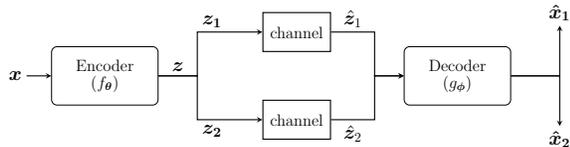


Fig. 6. Single decoder scheme with two layers.

A down side of the model described in the previous sections is the fact that a separate encoder and/or decoder needs to be trained for each layer. Here we try an alternative model that uses a single encoder and a single decoder for all transmissions, as described in Figure 6. This represents a considerable reduction in the algorithm’s complexity, both in memory and in processing, as the model size is constant regardless of the number of layers.

In order to retrieve information from partial codes, the decoder has to be trained for different code sizes. We achieve that by keeping a fixed loss function that compares inputs and outputs, while randomly varying the length of the code transmitted, at every batch. In practical terms, that meant creating a CNN model with fixed channel bandwidth $k = \sum_{i=1}^L k_i$, but randomly masking regions of the received message \hat{z} with zeros. In this way, the network could learn to specialize different regions of the code, using the initial parts to encode the main image content and the extra (often erased) parts for additional layers.

The results show that the performance of the single decoder scheme is surprisingly powerful, as can be seen in Figure 4b. The values are comparable to the multiple decoder case, making this scheme attractive.

VI. SUMMARY AND CONCLUSIONS

This work explored the use of deep learning based methods for the development of progressive JSCC strategies for image transmission. Building on recent results showing that artificial neural networks can be very effective in learning end-to-end JSCC algorithms, we explored whether the network can be extended to also learn successive refinement strategies, which would provide additional flexibility. To the best of our knowledge, no such hierarchical JSCC scheme has been previously developed and tested for practical information sources and channels.

We presented different strategies and models for progressive refinement - namely the use of multiple decoders, the transmission of residual images, and the use of a

single encoder. The results not only reproduce the effects observed in the previous work, such as impressive performance at low SNRs, limited bandwidth, and graceful degradation with test SNR, but also show the ability of neural networks in enabling progressive image transmission with almost no loss in the performance.

The best performance is obtained when a combination of one encoder and multiple receivers are trained jointly; however, alternative, less expensive strategies such as the communication of residuals instead of complete images and the use of a single decoder also showed comparable results, being viable options depending on the needs of the deployed system.

REFERENCES

- [1] K. R. Sloan and S. L. Tanimoto, “Progressive refinement of raster images,” *IEEE Transactions on Computers*, vol. 28, no. 11, pp. 871–874, 1979.
- [2] Y. Steinberg and N. Merhav, “On hierarchical joint source-channel coding,” in *International Symposium on Information Theory, 2004. ISIT 2004. Proceedings.*, Jun. 2004, pp. 365–365.
- [3] W. H. R. Equitz and T. M. Cover, “Successive refinement of information,” *IEEE Transactions on Information Theory*, vol. 37, no. 2, pp. 269–275, Mar. 1991.
- [4] E. Bourtsoulatzé, D. B. Kurka, and D. Gunduz, “Deep Joint Source-Channel Coding for Wireless Image Transmission,” *arXiv:1809.01733 [cs, eess, math, stat]*, Sep. 2018.
- [5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.
- [6] H. Kim, Y. Jiang, R. Rana, S. Kannan, S. Oh, and P. Viswanath, “Communication algorithms via deep learning,” in *Proc. of Int. Conf. on Learning Representations (ICLR)*, 2018.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] T. J. O’Shea, T. Erpek, and T. C. Clancy, “Deep learning based MIMO communications,” *arXiv:1707.07980 [cs.IT]*, 2017.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] R. Zarcone, D. Paiton, A. Anderson, J. Engel, H. S. P. Wong, and B. Olshausen, “Joint source-channel coding with neural networks for analog data compression and storage,” in *2018 Data Compression Conference*, March 2018, pp. 147–156.
- [15] 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.
- [16] 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.
- [17] 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.