Deep Joint Source-Channel Coding for Wireless Image Transmission

3 authors:

Eirina Bourtsoulatze
University College London
29 PUBLICATIONS   442 CITATIONS

SEE PROFILE

David Burth Kurka
Imperial College London
30 PUBLICATIONS   392 CITATIONS

SEE PROFILE

Deniz Gündüz
Imperial College London
455 PUBLICATIONS   7,213 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project   COPES - COnsumer-centric Privacy in smart Energy gridS View project

Project   2022 IEEE 96th Vehicular Technology Conference: VTC2022-Fall View project

Deep Joint Source-Channel Coding for Wireless Image Transmission

Eirina Bourtsoulatze, David Burth Kurka and Deniz Gündüz

Abstract—We propose a joint source and channel coding (JSCC) technique for wireless image transmission that does not rely on explicit codes for either compression or error correction; instead, it directly maps the image pixel values to the real/complex - valued channel input symbols. We parameterize the encoder and decoder functions by two convolutional neural networks (CNNs), which are trained jointly, and can be considered as an *autoencoder* with a non-trainable layer in the middle that represents the noisy communication channel. Our results show that the proposed deep JSCC scheme outperforms digital transmission concatenating JPEG or JPEG2000 compression with a capacity achieving channel code at low signal-to-noise ratio (SNR) and channel bandwidth values in the presence of additive white Gaussian noise. More strikingly, deep JSCC does not suffer from the "cliff effect", and it provides a graceful performance degradation as the channel SNR varies with respect to the training SNR. 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.

I. INTRODUCTION

Modern communication systems employ a two step encoding process for the transmission of image/video data (see Fig. 1a for an illustration): *(i)* the image/video data is first compressed with a source coding algorithm in order to get rid of the inherent redundancy, and to reduce the amount of transferred information; and *(ii)* the compressed bitstream is first encoded with an error correcting code, which enables resilient transmission against errors, and then modulated. Shannon's *separation theorem* proves that this two-step source and channel coding approach is optimal theoretically in the asymptotic limit of infinitely

E. Bourtsoulatze is with the Communications and Information Systems Group, Department of Electronic and Electrical Engineering, University College London, London, UK. D. Burth Kurka and D. Gunduz are with the Information Processing and Communications Laboratory, Department of Electrical and Electronic Engineering, Imperial College London, London, UK. Part of this work was done while the first author was with the Information Processing and Communications Laboratory, Imperial College London.

E-mails: `e.bourtsoulatze@ucl.ac.uk`, `d.kurka@imperial.ac.uk`, `d.gunduz@imperial.ac.uk`

long source and channel blocks [1]. While joint source and channel coding (JSCC) is known to outperform the separate approach [2], separate architecture is attractive for practical communication systems thanks to the modularity it provides. Moreover, highly efficient compression algorithms (e.g. JPEG, JPEG2000, WebP [3]) and near-optimal channel codes (e.g. LDPC, Turbo codes) are employed in practice to approach the theoretical limits. However, many emerging applications from the Internet-of-things to autonomous driving and to tactile Internet require transmission of image/video data under extreme latency, bandwidth and/or energy constraints, which preclude computationally demanding long-blocklength source and channel coding techniques.

We propose a JSCC technique for wireless image transmission that directly maps the image pixel values to the real/complex-valued channel input symbols. Inspired by the success of unsupervised deep learning (DL) methods, in particular the autoencoder architectures [4], [5], we design an end-to-end communication system, where 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*. Two channel models, the additive white Gaussian noise channel and the slow Rayleigh fading channel, are considered in this work due to their widespread adoption in representing realistic channel conditions. The proposed solution is readily extendable to other channel models, which can be represented as a non-trainable NN layer with a differentiable transfer function.

DL-based methods, and, particularly, autoencoders, have recently shown remarkable results in image compression, achieving or even surpassing the performance of state-of-the-art lossy compression algorithms [6]–[8]. 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. While common compression algorithms, such as JPEG, apply the same processing pipeline to all types of images (e.g., DCT transform, quantization and entropy coding in JPEG), the DL-based image compression algorithms learn the statistical characteristics from a large training dataset, and optimize the compression algorithm accordingly, without explicitly

specifying a transform or a code.

At the same time, the potential of DL has also been capitalized by researchers to design novel and efficient coding and modulation techniques in communications. In particular, the similarities between the autoencoder architecture and the digital communication systems have motivated significant research efforts in the direction of modelling end-to-end communication systems using the autoencoder architecture [9], [10]. Some examples of such designs include decoder design for existing channel codes [11], [12], blind channel equalization [13], learning physical layer signal representation for SISO [10] and MIMO [14] systems, OFDM systems [15], [16], and JSCC of text messages [17].

In this work, we leverage the recent success of DL methods in image compression and communication system design to propose a novel JSCC algorithm for image transmission over wireless communication channels. We consider both time-invariant and fading additive white Gaussian noise (AWGN) channels, and compare the performance of our algorithm to the state-of-the-art compression algorithms combined with capacity-achieving channel codes. 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 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: our algorithm exhibits graceful performance degradation when the channel conditions deteriorate. 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.

The rest of the paper is organized as follows. In Section II we introduce the system model, provide some background on the conventional wireless image transmission systems and its limitations, and motivate our novel approach. We introduce the proposed deep JSCC architecture in Section III. Section IV is dedicated to the evaluation of the performance of the proposed deep JSCC scheme, and its comparison with the conventional separate JSCC schemes over both static and fading AWGN channels. Finally, the paper is concluded in Section V.

II. BACKGROUND AND PROBLEM FORMULATION

We consider image transmission over a point-to-point wireless communication channel. The transmitter maps the input image $\boldsymbol{x} \in \mathbb{R}^n$ to a vector of complex-valued
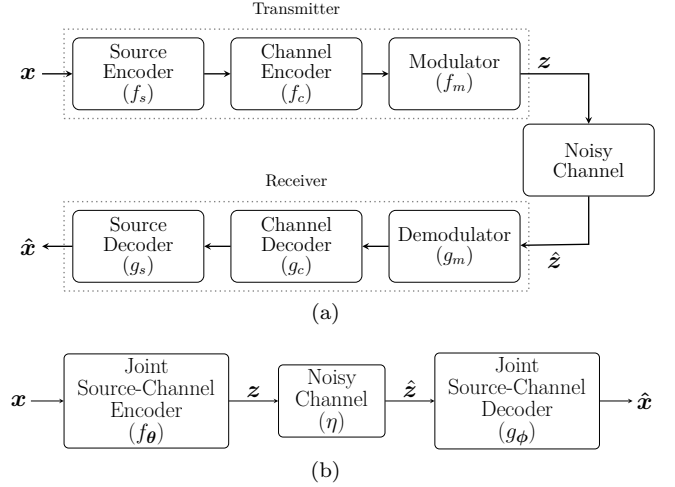

Fig. 1. Block diagram of the point-to-point image transmission system: (a) components of the conventional processing pipeline and (b) components of the proposed deep JSCC algorithm.

channel input symbols $\boldsymbol{z} \in \mathbb{C}^k$. 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*. Due to practical considerations in real-world communication systems, e.g., limited energy, interference, *etc.*, the output of the transmitter may be required to satisfy a certain power constraint, such as peak and/or average power constraints. The output signal $\boldsymbol{z}$ is then transmitted over the channel, which 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 $\hat{\boldsymbol{x}} \in \mathbb{R}^n$ of the original input image.

In conventional image transmission systems, depicted in Fig. 1a, the transmitter performs three consecutive independent steps in order to generate the signal $\boldsymbol{z}$ transmitted over the channel. First, the source redundancies are removed with a source encoder f_s, which is typically one of the commonly used compression methods (e.g., JPEG/JPEG2000, WebP). A channel code f_c (e.g., LDPC, Turbo code) is then applied to the compressed bitstream in order to protect it against the impairments introduced by the communication channel. Finally, the coded bitstream is modulated with a modulation scheme f_m (e.g., BPSK, 16-QAM) which maps the bits to complex-valued samples, which are then carried by the I and Q digital signal components over the communication link (the latter two components are often combined into a single coded-modulation step [18]).

The decoder inverts these operations in the reverse order. It first demodulates and maps the complex-valued channel output samples to a sequence of bits (or, log likelihood ratios) with a demodulation scheme g_m that

matches the modulator f_m. It then decodes the channel code with a channel decoding algorithm g_c, and finally provides an approximate reconstruction of the transmitted image from the (possibly corrupted) compressed bitstream by applying the appropriate decompression algorithm, g_s.

Though the above encoding process is highly optimized and widely adopted in image transmission systems [19], its performance may suffer severely when the channel conditions differ from those for which the system has been optimized. Although the source and channel codes can be designed separately, their rates are chosen jointly targeting a specific channel quality, i.e., assuming that a capacity achieving channel code can be employed, the compression rate is chosen to produce exactly the amount of data that can be reliably transmitted over the channel. However, when the experienced channel condition is worse than the one for which the code rates are chosen, meaning that 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.

Similarly, the separate design cannot benefit from improved channel conditions either; that is, once the source and channel coding rates are fixed, no matter how good the channel is, the reconstruction quality remains the same as long as the channel capacity is above the target rate. These two characteristics are known as the "cliff effect". Various joint source-channel coding schemes have been proposed in the literature to overcome the "cliff effect" [20], [21], and to obtain graceful degradation of the end-to-end signal quality with channel SNR, which typically combine multi-layer digital codes with multi-layer compression for unequal error protection.

In this paper we take a radically different approach, and leverage the properties of uncoded transmission [22], [23] by directly mapping the real pixel values to real/complex-valued samples transmitted over the communication channel. 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, directly maps the pixel values to channel inputs.

III. 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 [6], [8], as well as by the first promising results in the design of end-to-end communication systems using autoencoder architectures [9], [10].

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 $\boldsymbol{\theta}$. The encoder CNN comprises a series of convolutional layers followed by a fully connected layer and a normalization layer. The convolutional layers extract the image features which are subsequently combined by the fully connected layer to form the channel input samples. The output $\tilde{\boldsymbol{z}} \in \mathbb{C}^k$ of the fully connected layer is normalized according to:

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

where $\tilde{\boldsymbol{z}}^*$ is the conjugate transpose of $\tilde{\boldsymbol{z}}$, such that the 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 be able to optimize the communication system in Fig. 1b in an end-to-end manner, the communication channel must be incorporated into the overall NN architecture. We model the communication channel as a series of non-trainable layers which are represented by the transfer function $\hat{\boldsymbol{z}} = \eta(\boldsymbol{z})$. We consider two widely used channel models: *(i)* the additive white Gaussian noise channel, and *(ii)* the slow fading channel. The transfer function of the Gaussian channel is $\eta_n(\boldsymbol{z}) = \boldsymbol{z} + \boldsymbol{n}$, where the vector $\boldsymbol{n} \in \mathbb{C}^k$ consists of independent identically distributed (i.i.d.) samples from a circularly symmetric complex Gaussian distribution, i.e., $\boldsymbol{n} \sim \mathcal{CN}(0, N_0\boldsymbol{I}_k)$. In the case of slow fading channel, we adopt the commonly used Rayleigh slow fading model. The multiplicative effect of the channel gain on the transmitted signal is captured by the channel transfer function $\eta_h(\boldsymbol{z}) = \boldsymbol{H}\boldsymbol{z}$, where $\boldsymbol{H}$ is a diagonal matrix with circularly symmetric complex Gaussian random vector $\boldsymbol{h} \sim \mathcal{CN}(0, H_c\boldsymbol{I}_k)$ on the main diagonal. The joint effect of channel fading and Gaussian noise can be modelled by the composition of the transfer functions η_h and η_n: $\eta(\boldsymbol{z}) = \eta_n(\eta_h(\boldsymbol{z})) = \boldsymbol{H}\boldsymbol{z} + \boldsymbol{n}$. 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 comprises a joint source-channel decoder. The decoder maps the corrupted complex-valued signal $\hat{\boldsymbol{z}} = \eta(\boldsymbol{z}) \in \mathbb{C}^k$ to an estimation of the original input $\hat{\boldsymbol{x}} \in \mathbb{R}^n$ 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 received (and possibly corrupted) signal $\hat{\boldsymbol{z}}$ through a fully connected neural layer to obtain the image features. It then applies a series of
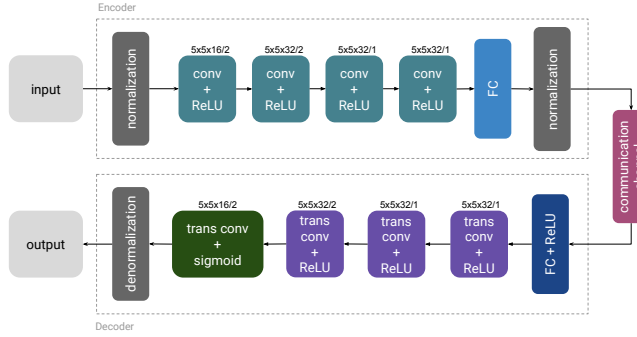
Fig. 2. Encoder and decoder NN architectures used in the implementation of the proposed deep JSCC scheme.

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:

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

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 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.

IV. EVALUATION

To demonstrate the potential of our proposed deep JSCC scheme, we use the NN architecture depicted in Fig. 2. At the encoder, the normalization layer is followed by four convolutional layers and a fully connected layer. Since the statistics of the input data are generally not known at the decoder, the input images are normalized by the maximum pixel value 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 values of the hyperparameters F, K and S used in our experiments are given in Fig. 2. ReLU activation function is applied to the output of all convolutional layers. The output of the last convolutional layer is fed into a fully connected (FC) layer, which transforms the image features into the encoded representation. The output of the fully connected layer is of size $2k$ for complex-valued channel input samples and k for real-valued channel input samples. The fully connected 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 fully connected layer of input size $2k$, if $\hat{\boldsymbol{z}} \in \mathbb{C}^k$, or k, if $\hat{\boldsymbol{z}} \in \mathbb{R}^k$, and then into the transpose convolutional layers, which progressively transform the corrupted image features into an estimation 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 output of the fully connected layer and the first three transpose convolutional layers of the decoder are passed through a ReLU activation function, while a sigmoid nonlinearity is applied to the output of the last transpose convolutional layer in order to produce values in the $[0, 1]$ range. Finally, a denormalization layer multiplies the output values by 255, and clips them in order to generate pixel values within the $[0, 255]$ range.

The above architecture is implemented in Tensorflow [24]. The training data consists of the $N = 50000$ CIFAR-10 32×32 training images [25] combined with random realizations of the channel under consideration. We use the Adam optimization framework [26], 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

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} d(\boldsymbol{x}_i, \hat{\boldsymbol{x}}_i), \qquad (3)$$

where $d(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{1}{n}||\boldsymbol{x} - \hat{\boldsymbol{x}}||^2$ is the mean squared-error distortion.

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 $\eta = \eta_n$. Without loss of generality, we assume that the channel input samples as well as the noise samples are real-valued, i.e. $\text{Im}\{\boldsymbol{z}\} = \text{Im}\{\boldsymbol{n}\} = \boldsymbol{0}$ and $\text{Re}\{\boldsymbol{n}\} \sim \mathcal{N}(0, N_0 \boldsymbol{I}_k)$. We set the average power constraint to $P = 1$, and vary the channel SNR by varying the noise variance N_0. The channel SNR is computed as:

$$\text{SNR} = 10 \log_{10} \frac{P}{N_0} \text{ dB}. \qquad (4)$$

The performance of the deep JSCC algorithm is quantified in terms of the PSNR of the reconstructed images at the output of the decoder, defined as follows:

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\text{MSE}} \text{ dB}. \qquad (5)$$

where $\text{MSE} = d(\boldsymbol{x}, \hat{\boldsymbol{x}})$.

We compare the proposed deep JSCC algorithm with a digital transmission scheme, which employs JPEG or JPEG2000 for compression followed by a capacity-achieving channel coding and modulation scheme. We first compute the maximum number of bits per source
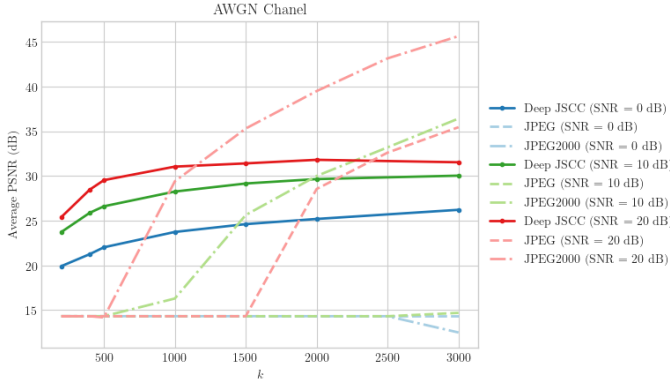
Fig. 3. Performance of the learned JSCC algorithm on test images over an AWGN channel with respect to the channel bandwidth, k, for different SNR values. For each case, the same target SNR value is used in training and evaluation.

sample that can be transmitted reliably, according to the Shannon's separation theorem:

$$R = \frac{k}{n}C, \qquad (6)$$

where $C = \frac{1}{2}\log_2(1+\text{SNR})$ for a real AWGN channel and $C = \log_2(1+\text{SNR})$ for a complex AWGN channel. Since JPEG and JPEG2000 cannot compress the image data at an arbitrarily low bitrate, we also compute the minimum bitrate value R_{min} beyond which compression results in complete loss of information and the original image cannot be reconstructed. For all bitrate values R computed using Eq. (6) which are below the minimum bitrate value R_{min}, we assume that the image is reconstructed to the mean value of all the pixels. For bitrate values R above R_{min}, we compress the images at the largest rate R' that satisfies $R' \leq R$, since again it is not always possible to achieve an arbitrary target bitrate R with JPEG or JPEG2000 compression software.

Fig. 3 illustrates the performance comparison between the proposed deep JSCC algorithm and the digital schemes as a function of the channel bandwidth k in different SNR regimes. We note that the threshold behavior of the digital schemes in the figures are not due to the "cliff effect", as we already plot here the performance of the digital schemes optimized for the corresponding channel SNR. The initial flat part of these curves is due to the fact that JPEG and JPEG2000 completely break down in this region, i.e., the compression rate is below the minimum required number of bits per pixel, R_{min}, to recover a meaningful reconstruction of the images.

We observe that, when the channel bandwidth is limited, i.e., for $k \in [200, 1000]$, the performance of the proposed deep JSCC scheme is considerably above the one that can be achieved by JPEG and JPEG2000 even assuming that reliable transmission at channel capacity is

possible[1]. 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 believe that the saturation of the proposed deep JSCC scheme in the large channel bandwidth regime is due to the limited capability of the autoencoder architecture, which can be improved through various techniques that have been introduced in the DNN-based image compression literature, e.g., [8], [28].

We next study the robustness of the proposed deep JSCC scheme to variations in channel conditions. Figs. 4a and 4b illustrate the average PSNR of the reconstructed images versus the SNR of the additive white Gaussian noise channel for two different values of channel bandwidth, k. Each curve in Figs. 4a and 4b is generated by training our end-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 10000 test CIFAR-10 images for varying SNR values, denoted as SNR_{test}. In other words, each curve represents the performance of the proposed JSCC scheme optimized for channel SNR equal to SNR_{train}, and deployed in different channel conditions with SNR equal to SNR_{test}. These results provide an insight into the performance of the proposed algorithm when the channel conditions are different from those for which the end-to-end system is optimized. We can observe that for $\text{SNR}_{test} < \text{SNR}_{train}$, i.e., when the channel conditions are worse than those for which the encoder/decoder have been optimized, our deep JSCC algorithm does not suffer from the "cliff effect" observed in digital systems. Unlike digital systems, where the quality of the decoded signal drops sharply when SNR_{test} drops below a critical value close to the SNR_{train}, the 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 [20], [22], and is attributed to the capability of the autoencoder to map similar images/features to nearby points in the channel input signal space; thus, with decreasing SNR_{test} the decoder can still obtain a reconstruction of the original image.

On the other hand, when SNR_{test} increases above SNR_{train}, we observe initially a gradual improvement in the 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

---

[1]While near capacity-achieving channel codes exist for the AWGN channel, these typically require very large blocklength. It is known that the achievable rates guaranteeing low block error probability for the blocklengths considered here are below the capacity [27]. Therefore, the curves for the digital schemes in Fig. 3 are typically not achievable, and can serve as upper bounds.

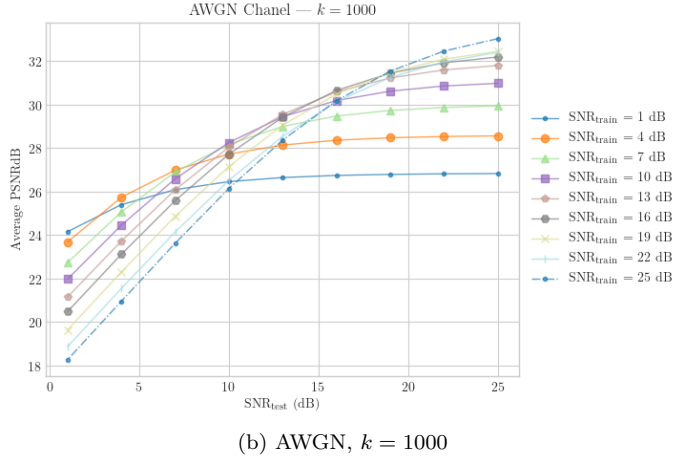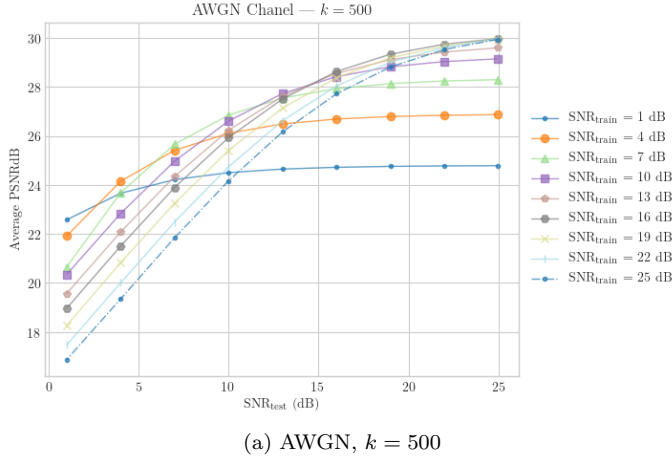(a) AWGN, $k = 500$

(b) AWGN, $k = 1000$

Fig. 4. Performance of the learned JSCC algorithm on test images with respect to the channel SNR over an AWGN channel. Each curve is obtained by training the encoder/decoder network for a particular channel SNR value.
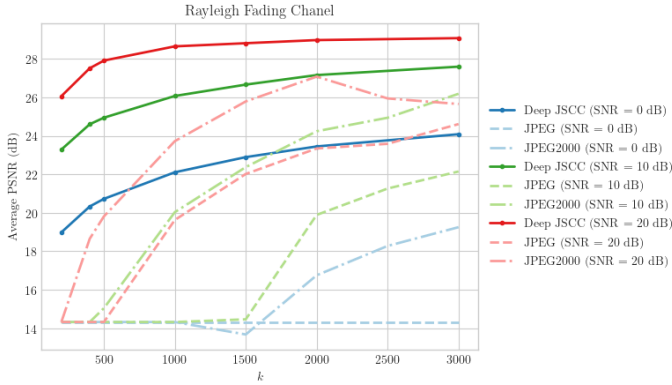

Fig. 5. Performance of the learned JSCC algorithm on test images over a slow Rayleigh fading channel with respect to the channel bandwidth, k, for different SNR values. For each case, the same target SNR value is used in training and evaluation.

implicitly decided during the training phase for the target value SNR_{train}. It is worth noting that performance saturation does not occur at $\text{SNR}_{test} = \text{SNR}_{train}$ as in digital image/video transmission systems [23], but at $\text{SNR}_{test} > \text{SNR}_{train}$. This behavior indicates that the proposed JSCC scheme determines an implicit trade-off between the amount of error protection and compression, which does not necessarily target an error-free transmission when the system operates at $\text{SNR}_{test} = \text{SNR}_{train}$. We also note that when the encoder/decoder are optimized for very high SNR_{train}, and $\text{SNR}_{test} > \text{SNR}_{train}$, the system boils down to an ordinary autoencoder, and its performance is solely limited by the degree-of-freedom imposed by the channel bandwidth k, i.e., the dimension of the bottleneck layer of the autoencoder.

We also study the performance of our deep JSCC scheme under the assumption of a slow Rayleigh fading channel

with additive white Gaussian noise. 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 $\eta(\boldsymbol{z}) = \text{diag}(\boldsymbol{h})\boldsymbol{z} + \boldsymbol{n}$, 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 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

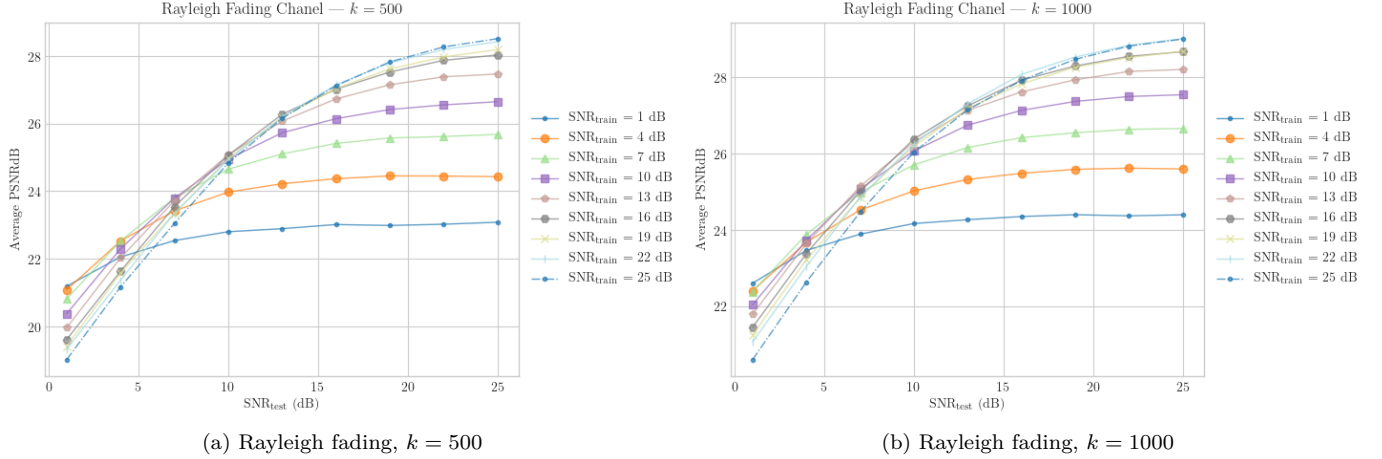(a) Rayleigh fading, $k = 500$ (b) Rayleigh fading, $k = 1000$

Fig. 6. Performance of the learned JSCC algorithm on test images with respect to the average channel SNR over an AWGN Rayleigh fading channel. Each curve is obtained by training the encoder/decoder network for a particular channel SNR value.

varying channel states.

We also illustrate the robustness of the proposed deep JSCC scheme to variations of the average channel SNR in Figs. 6a and 6b. We observe that, while the performance of the deep JSCC scheme drops compared to the static AWGN channel, the difference is not significant, despite the lack of channel state information. This suggests that the network learns to estimate the channel state, and adapts the decoder accordingly; that is, the proposed deep JSCC scheme combines not only source coding, channel coding, and modulation, but also channel estimation, into one single component whose parameters are learned through training.

Finally, in Fig. 7 we present some examples of the original and reconstructed images by the deep JSCC scheme. These figures are obtained for AWGN channel at SNR = 10 dB. We observe that a reasonable quality image can be reconstructed even for very small channel bandwidths, but the quality does not improve significantly going from $k = 1000$ to $k = 2000$. As stated earlier, we believe that this performance saturation can be attributed to the limitations of the employed autoencoder structure.

V. Conclusions and Future Work

We have proposed a novel deep JSCC architecture for image transmission over wireless channels. In this architecture, the encoder maps the input image directly to channel inputs. The encoder and the decoder functions are modeled as complementary DNNs, and trained jointly on the dataset to minimize the average MSE of the reconstructed image. We have compared the performance of this deep JSCC scheme with conventional separation-based digital transmission schemes, considering state-of-the-art image compression algorithms followed by capacity achieving channel codes. 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.

In the case of DL-based JSCC, the encoder and decoder networks learn not only to communicate reliably over the channel (as in [10], [12], but also to compress the images efficiently. For a perfect channel with no noise, if the source bandwidth is greater than the channel bandwidth, i.e., $n > k$, the encoder-decoder NN pair is equivalent to an *undercomplete autoencoder* [5], which effectively learns the most salient features of the training dataset. However, in the case of a noisy channel, simply learning a good low-dimensional representation of the input is not sufficient. The network should also learn to map the salient features to nearby representations so that similar images can be reconstructed despite the presence of noise. We also note that, the resilience to channel noise acts as a sort of a regularizer for the autoencoder. For example, when there is no channel noise, if the channel bandwidth is larger than the source bandwidth, i.e., $n < k$, we obtain an *overcomplete autoencoder*, which can simply learn to replicate the image. However, when there is channel noise, even an overcomplete autoencoder learns a non-trivial mapping that is resilient to channel noise, similarly to denoising autoencoders.

The next step in improving the performance of the deep JSCC scheme is to exploit more advanced NN architectures in the autoencoder that have been shown to improve the compression performance [8], [28]. We will also explore

Fig. 7. Examples of reconstructed images for different channel bandwidth values. Note how color and texture are enhanced as the channel bandwidth increases.

the performance of the system for non-Gaussian channels as well as for channels with memory, for which we do not have capacity-approaching channel codes. We expect that the benefits of the proposed NN-based JSCC scheme will be more evident in these non-ideal settings.

REFERENCES

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

[2] F. Zhai, Y. Eisenberg, and A. K. Katsaggelos, "Joint source-channel coding for video communications," in *Handbook of Image and Video Processing*, 2nd ed., A. Bovik, Ed. Burlington: Academic Press, 2005.

[3] Google, "WebP compression study." [Online]. Available: https://developers.google.com/speed/webp/docs/webp_study

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

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] H. Kim *et al.*, "Communication algorithms via deep learning," in *Proc. of Int. Conf. on Learning Representations (ICLR)*, 2018.

[12] E. Nachmani *et al.*, "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.

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

[18] A. G. Fabregas, A. Martinez, and G. Caire, "Bit-interleaved coded modulation," *Foundations and Trends in Communications and Information Theory*, vol. 5, no. 1-2, pp. 1–153, 2008.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

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

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

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

[27] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[28] N. Johnston *et al.*, "Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.