Deep Joint Source Channel Coding via Attention for Wireless Image Transmission

Haoze Chang, Lin Ma(✉), and Xuedong Wang

School of Electronics and Information Engineering,
Harbin Institute of Technology, Harbin, China
malin@hit.edu.cn

Abstract. In digital communication, efficiently transmitting image and video data through constrained channels remains challenging nowadays. Traditional methods using separate source and channel coding often fail in dynamic environments. In this paper, we introduce a novel deep learning based (DL) attention joint source channel coding (AttenJSCC) approach, which enhances robustness and efficiency in wireless image transmissions. By integrating source and channel coding into a unified framework and incorporating our Enhanced Attention Feature (EAF) modules and the ECA attention mechanism, our method outperforms some of the existing JSCC techniques, especially in low SNR conditions. Our framework not only overcomes the limitations of current technologies but also reduces the storage and computational needs on edge devices, facilitating more efficient real time communication.

Keywords: JSCC · Deep Learning · Wireless Image Transmission · Attention Mechanisms

1 Introduction

With the rapid advancement of modern computer science and the growing demands of users, the rise of Internet of Things applications is pushing the limits for transmitting image/video data under the strict conditions of latency, bandwidth, and energy consumption [1]. Contemporary communication frameworks employ a dual phase encoding method for distributing image/video content, known as source coding and channel coding [2], as illustrated in Fig. 1. Although this encoding process is highly optimized and widely adopted in image transmission systems, its performance can be significantly compromised when the channel conditions deviate from those for which the system was optimized. Such performance degradation is known as the *cliff effect*, which has been seen as a common problem in the digital communication scheme.

To address the limitations posed by traditional separate source channel coding schemes, several deep learning based joint source channel coding (JSCC) methods have been introduced [3–5]. Due to the strong encoding and decoding

H.-H. Chen and W. Meng (Eds.): WiSATS 2024, LNICST 605, pp. 312–323, 2025.
https://doi.org/10.1007/978-3-031-86196-3_27

ability of deep learning architecture, the joint source channel coding architecture can help capture the most important features in the original image. The feature vector is mapped to the complex value channel sample. The DeepJSCC [6] scheme, specifically tailored for wireless image transmission, has demonstrated significant potential by directly mapping image pixels to complex value channel input symbols. Building on the pioneering work of DeepJSCC, numerous related JSCC projects have been developed. For instance, DeepJSCC-f [7] leverages feedback from the receiver, utilizing the feedback information through deploying a decoder at the transmitter end to modify the outgoing data based on the received information. To align with modern hardware capabilities, DeepJSCC-Q [8] has been proposed to map complex value signals into constellation signals with fixed positions, thereby circumventing the quantization process. However, some of the approaches mentioned above require training across multiple channel SNRs to adapt to diverse channel conditions. Such designs necessitate increased storage capacity on edge devices and precise channel estimation prior to image transmission. Although some models are trained within a predefined SNR range to accommodate channel variability, their performance is often compromised under low SNR conditions, leading to diminished effectiveness.

In this study, we leverage recent advancements in deep learning methodologies within the realms of image compression and communication systems to introduce a cutting-edge joint source channel coding algorithm designed for image transmission across wireless communication channels. We proposed a new module based on the attention mechanism to recalibrate the weight of feature map and thus amplify the core features under different channel SNRs.

The remainder of the paper is organized as follows: Sect. 2 discusses the theoretical foundation and the model overview of our proposed JSCC algorithm. Section 3 presents the proposed Enhanced Attention Feature module and gives a detailed description on the model architecture. Section 4 gives a detailed explanation on experimental setup, including the datasets used, training procedures, and the evaluation metrics. Finally, the conclusions and future research directions are summarized in Sect. 5.

2 System Overview

In this section, we will give a detailed introduction of the proposed end-to-end image wireless transmission. In the traditional transmission method, the message is firstly encoded to remove the redundancies in the message and thus reduce the bits needed to be sent. The channel coding is done next to add protective bits to detect and correct possible mistakes. The process can be represented as:

$$\mathbf{b} = \mathbf{c}_\beta(\mathbf{s}_\alpha(M)) \tag{1}$$

where $\mathbf{c}_\beta(\cdot)$ denotes the channel coding algorithm β, and $\mathbf{s}_\alpha(\cdot)$ denotes the source coding algorithm α. M denotes the message to be sent. The encoded information is further processed through modulation and frequency up conversion before being transmitted in the high frequency band. At the receiver end, the received

bits are first frequency shifted to the base band and the rest process is done reversely, that is, the protective channel coding bits are first removed to restore the original message after source coding. The sent message is restored through the source decoding. The process can be presented as:

$$\hat{M} = \mathbf{s}_{\alpha}^{-1}(\mathbf{c}_{\beta}^{-1}(\hat{\mathbf{b}})) \tag{2}$$

where $\mathbf{c}_{\beta}^{-1}(\cdot)$ denotes the decoding algorithm based on the channel coding algorithm β and $\mathbf{s}_{\alpha}^{-1}(\cdot)$ denotes the source decoding algorithm based on the source coding algorithm α. The $\hat{M}$ denotes the message the receiver desired. We use the notation $\hat{b}$ rather than b in the formula 2 to denote the effect of channel noise. The overall process can be expressed in Fig. 1.

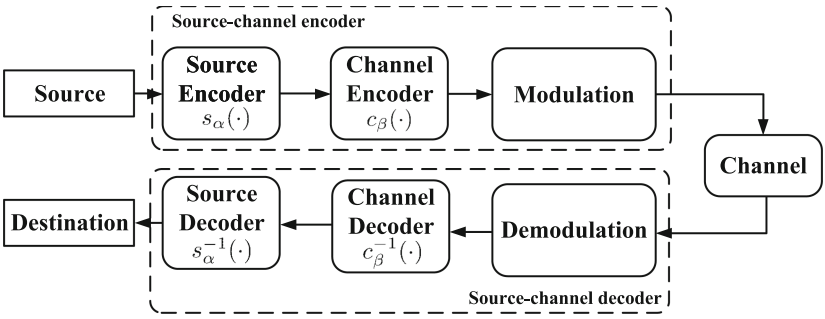


Fig. 1. Traditional digital communication block diagram. Source will pass through source encoder, channel encoder and modulator to generate channel sample.

In the deep learning based image wireless transmission, the architecture of our designed system is made up of two parts namely encoder and decoder, which respectively equals the $\mathbf{c}_{\beta}(\mathbf{s}_{\alpha}(\cdot))$and the $\mathbf{s}_{\alpha}^{-1}(\mathbf{c}_{\beta}^{-1}(\cdot))$ in the formula 1 and formula 2. In the deep learning based communication system, the encoder and decoder is usually implemented using the deep learning neural network. Here in our work, we use the classic and practical CNN as the backbone. The encoder captures deep features embedded in the input image, namely $\mathbf{f} = \mathbf{E}(I)$, where I denotes the input image with a shape of $\mathbf{I} \in \mathbb{R}^{h \times w \times 3}$. The notation $\mathbf{E}$ denotes the encoder and the notation $\mathbf{f}$ denotes the output feature which has a shape $\mathbf{f} \in \mathbb{R}^{h' \times w' \times n}$. The total length of the vector $\mathbf{f}$ is $2k$. The $2k$ units in the $\mathbf{f}$ will be combined in groups of two to form the final channel sample $z \in \mathbb{C}^k$.

The output feature tensor will be passed through the channel as $\hat{z} = \mathbf{C}(z)$. The $\mathbf{C}(\cdot)$ refers to the communication channel. We rewrite the noising progress as $\hat{z} = z + n$ considering the AWGN channel used in our design, where $n \sim \mathcal{CN}(0, \sigma^2 \mathbf{I}_{k \times k})$ is a complex Gaussian vector with covariance matrix $\mathbf{I}_{k \times k}$ as a $k \times k$ identity matrix [9].

At the receiver end, the receiver takes the received tensor $\hat{z}$ and input it to the decoder side to restore the original image as $\bar{\mathbf{I}} = \mathbf{D}(\hat{z})$ which the $\mathbf{D}(\cdot)$ refers

to the decoder deployed on the receiver side. The entire transmission process can be written as:

$$\bar{I} = \mathbf{D}(\mathbf{C}(\mathbf{E}(I))) \tag{3}$$

The optimal parameters of the model can be get through optimizing the loss function through gradient descent while the optimization progress is written as:

$$\theta_{model} = \arg\min_{\theta} \mathcal{L}(I, \mathbf{D}(\mathbf{C}(\mathbf{E}(I)))) \tag{4}$$

We select the MSE loss to evaluate the distortion between the two images which is written as:

$$MSE = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \|(a_{ij} - b_{ij})\|^2 \tag{5}$$

We assume that both transmitter and receiver can estimate the channel noise power σ^2 and define SNR as:

$$SNR = 10 \log_{10}(\frac{P}{\sigma^2}) dB \tag{6}$$

Without loss of generality, we set the P as $P = 1$. For fair comparison, we define the *bandwidth compression ratio* as:

$$\gamma = \frac{k}{H \times W \times C} \tag{7}$$

To better evaluate the performance of the proposed scheme, We set the power constraint as $\frac{1}{k}\mathbb{E}[\hat{z}^* \hat{z}] \leq P$. The final output tensor is reshaped to form a tensor with the shape as $z \in 1 \times 1 \times 2k$. The real numbers of the output tensor is passed through a normalization layer to force the output tensor to satisfy the constraint of power following the equation $\hat{z} = \sqrt{kP}\frac{z}{\sqrt{z^* z}}$ in which the z^* denotes the conjugate transpose of z. The output of the normalization layer is combined into k complex value channel input samples and forms the encoded signal representation, which is transmitted over the channel.

3 Attention JSCC Method

In this section, we will give a brief introduction on the deep learning based joint source channel coding used in our system. Our system employs the typical encoder decoder architecture which can be deployed separately as deploying the encoder in the edge devices such as cell phones and deploying the decoder on the server end.

3.1 Encoder and Decoder

In this section, we provide an overview of the AttenJSCC system used in our system, depicted in Fig. 2. The architecture comprises an encoder that compresses the input image into a feature vector for transmission and a decoder that reconstructs the image from this feature vector. The encoder consists of four main components: the encoding block, encoding Res-block, the Enhanced Attention Feature (EAF) module, and the ECA attention block, with the decoder featuring symmetrical counterparts.

The encoding block includes a 2D convolution layer, generalized divisive normalization (GDN) [10] and a PReLU activation function, with the latter chosen for its adaptability to noised data due to a learnable parameter α when $x \leq 0$. The encoding Res-block [11], designed to mitigate degradation in deep learning, comprises two convolution layers. The input tensor is first padded to proper size and then normalized using a GDN layer and activated via PReLU. The subsequent layer repeats this process, excluding the final PReLU activation. An optional dropout layer is incorporated to prevent overfit. The ECA attention mechanism [12], known for its simplicity and effectiveness, has been used in multiple networks. The ECA attention mechanism pools the input tensor along the channel dimension to form a factor vector, which is then convoluted to scale the input tensor in the channel dimension. The scaled tensor is residual connected to the input tensor to form the encoding Res-block output tensor.

This tensor then enters the proposed EAF module which adjusts channel weights to enhance features and reduce noise impact through an attention mechanism. Received tensor will be flattened to add restore the height and width dimension. In the decoder, the Transposed Conv2D and the inverse GDN effectively reverse the encoding process. The rest parts in the decoding module remains symmetric as the encoder.

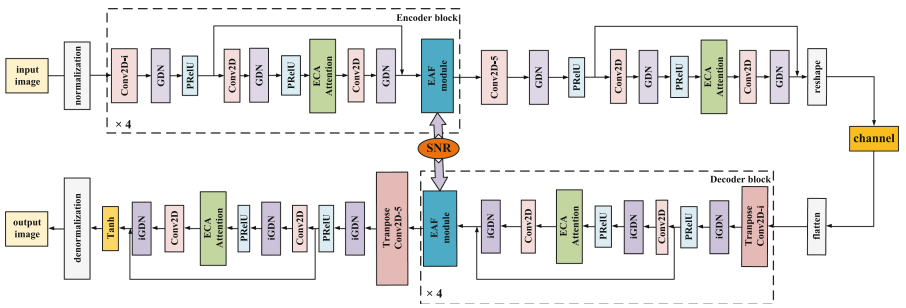


Fig. 2. Proposed model overview. We stack the Conv2D PReLU function, Res-block, GDN normalization and EAF module together as the encoder block, the final block doesn't include EAF module. '-i' denotes the i-th module and correspond to the parameter in the following section.

3.2 Enhanced Attention Feature Module

In this section, we introduce the Enhanced Attention Feature (EAF) Module used in our architecture. Previous approaches [13] used an Attention Feature (AF) module [14] that concatenated the SNR and feature vector to output a reweighed feature vector, aiming to mitigate noise effects. However, we suppose that a separation in the reweighting process, allowing the network to distinctly enhance image features and diminish noise effects. To better restore the original image, the weights of core feature maps should be amplified first and the amplified feature vector should be recalibrated with SNR afterwards.

Figure 3 depicts the EAF module's structure. It starts with global average pooling across the channel dimension of the input tensor to derive a factor tensor that reflects the weight of features across different channels. This tensor undergoes transformation through a linear layer and a PReLU function [15], repeated twice to refine the feature weighting factor. Subsequently, this factor tensor is merged(concatenated) with an SNR value to form an SNR factor vector, which is processed through another linear layer and a Sigmoid function to produce the final scaling weights. These weights are then applied element wise multiplication to the original tensor to produce the scaled tensor, effectively adjusting the image data in response to varying SNR levels. The performance of our design is evaluated in the following section.

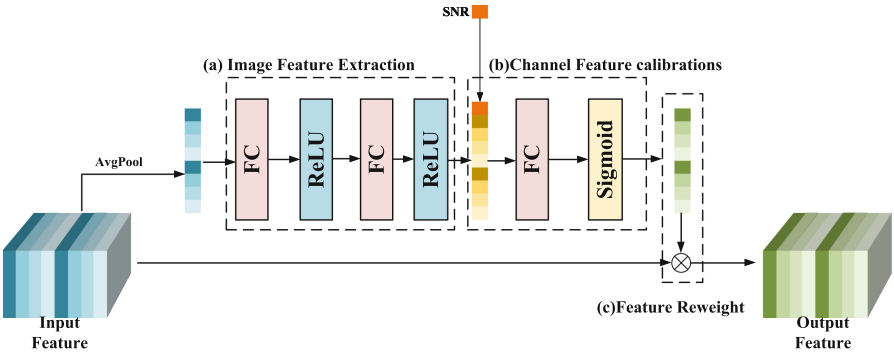


Fig. 3. Proposed **E**nhanced **A**ttention **F**eature Module. EAF module contains three parts: (a) Image Feature Extraction, (b) Channel Feature Calibrations, (c) Feature Reweight.

4 Implementation and Performance Evaluation

4.1 Experiment Setup

We conducted multiple experiments to evaluate our proposed model's effectiveness using the CIFAR-10 dataset, which contains 60,000 32×32 training images. We split the dataset into training and test sets at a $5:1$ ratio. The initial learning rate was set at 10^{-3} and was reduced by 0.1 every 400 epochs, with training

capped at 2000 epochs. We incorporated early stopping to halt training if the validation loss did not decrease by at least 0.00005 over 50 epochs. The training used a mini batch size of 64, and performance evaluation involved transmitting each image 50 times to counteract channel randomness.

Table 1. Encoder and Decoder parameters

Operation	Parameters	Operation	Parameters
conv2D-1	$9 \times 9 \times 256 \times 2 \times 4$	transconv2D-1	$5 \times 5 \times 256 \times 1 \times 2$
conv2D-2	$5 \times 5 \times 256 \times 2 \times 2$	transconv2D-2	$5 \times 5 \times 256 \times 1 \times 2$
conv2D-3	$5 \times 5 \times 256 \times 1 \times 2$	transconv2D-3	$5 \times 5 \times 256 \times 1 \times 2$
conv2D-4	$5 \times 5 \times 256 \times 1 \times 2$	transconv2D-4	$5 \times 5 \times 256 \times 2 \times 2$
conv2D-5	$5 \times 5 \times 256 \times 1 \times 2$	transconv2D-5	$8 \times 8 \times 3 \times 2 \times 2$

This setup was implemented in PyTorch, using the Adam optimizer [16] and MSE loss. Input images were normalized from $[0, 1]$ to $[-1, 1]$ by dividing by the maximum pixel value of 255 and normalization. The GDN and iGDN layers used a fixed reduction parameter of 16. The channel configuration was set as $[3, 256, 256, 256, 256, C]$, with the inverse configuration in the decoder. The C denotes the final output channel. The output channel will be set as 8 for the bandwidth ratio $\gamma = \frac{1}{12}$. The detailed parameters used in our design is listed in Table 1. The notation $K \times K \times F \times S \times P$ denotes a Conv2D/TransConv2D layer with F filters with kernel size $K \times K$ and stride S which pads the input tensor with P. For the 2D convolution layer used in the Res-block, we set the kernel size of the filter as 3.

4.2 Performance Evaluation

We first investigate the performance of our proposed deep AttenJSCC algorithm in the AWGN setting. We change the SNR used in the testing scenario and use PSNR [17] as the metric to evaluate the performance of the proposed scheme. Figure 4 illustrates the performance of our proposed scheme under different testing scenario while the compression is set to $\frac{1}{12}$. For low SNR regime, performance has surpassed some of the most powerful communication schemes in prior work. Our proposed scheme achieved an amazing performance and reached a PSNR of 22.97 dB when the channel situation is extremely limited (SNR $= -2$ dB, indicating that the power of noise has surpassed the power of signal). When the AttenJSCC is tested on 3dB, the performance is even on par with the some of the DL based methods trained on medium and high SNR region thus exhibiting stronger noise resistance capability. For high SNR regime, our work remains competitive as the results illustrate that our scheme outperforms prior work.

Figure 5 illustrates the performance of our proposed scheme under different testing scenario while the compression is set to $\frac{1}{6}$. Our work remains competitive under the whole SNR range. In challenging low SNR scenarios, performance

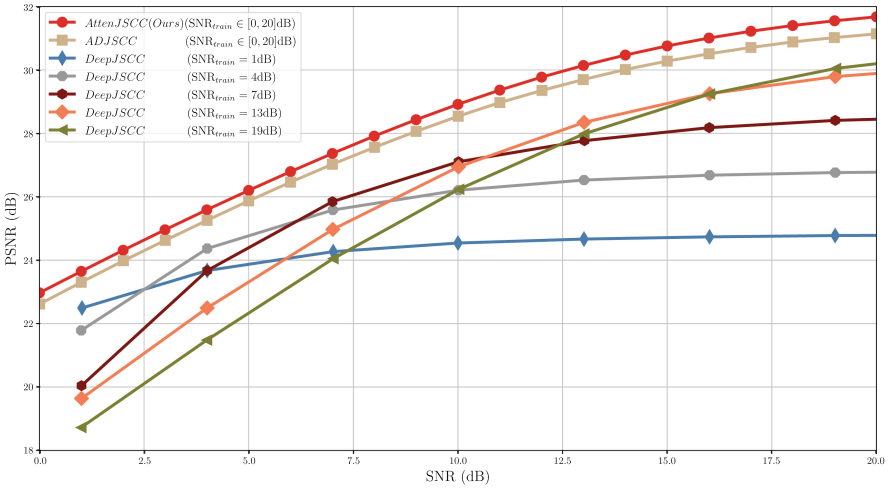


Fig. 4. Performance under different test SNR on CIFAR-10 test images while R = 1/12. The curve of AttenJSCC is trained under the uniform distribution of SNR from 0dB to 20 dB. Each curve of DeepJSCC is trained at a specific SNR.

of our work falls within a moderate range when compared to the DeepJSCC-f. However, we want to emphasize that our work has more advantages and is easier to implement. First, our work is trained on a target SNR range rather on specific target SNR value such that the edge devices only need to store one set of parameters rather than multiple sets. Second, our work is made up of simple encoder decoder architecture rather than complicated architecture with feedback. When compared with ADJSCC, the performance on higher SNR regime generally resides within a moderate spectrum. Our architecture demonstrates stronger image recovery capabilities in low SNR environments thus we believe that it may be prone to overfitting in high SNR regions. Such performance degradation can be seen as reasonable tradeoff for the performance improve under low SNR.

Also, to test our architecture's performance on larger images, we implemented extensive experiments on Kodak [18] images, which has 24 high quality images with 768×512 resolutions and is frequently used as a standard dataset for evaluating image compression performance. The performance is compared with DeepJSCC under different scenarios and the result is shown in Fig. 6. We trained our model on COCO2017 dataset [19] until convergence. COCO2017 is a high quality and high resolution image dataset with elaborate notations and contains

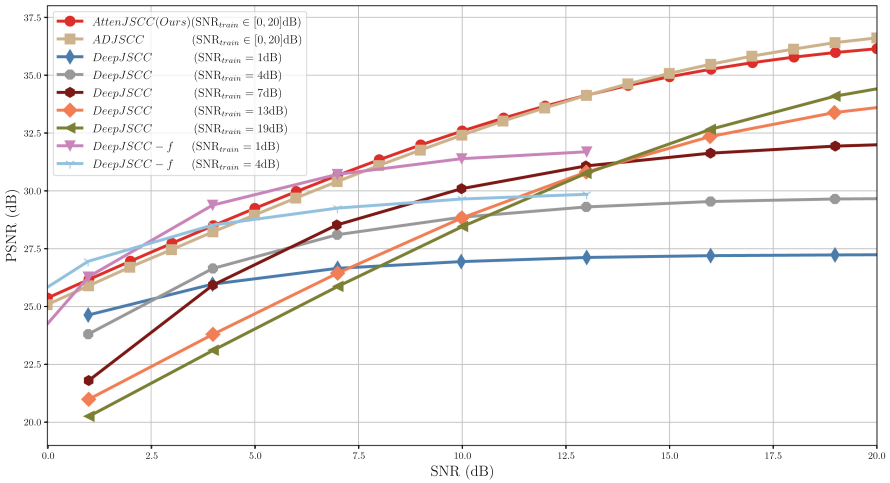


Fig. 5. Performance under different test SNR on CIFAR-10 test images while R =1/6.

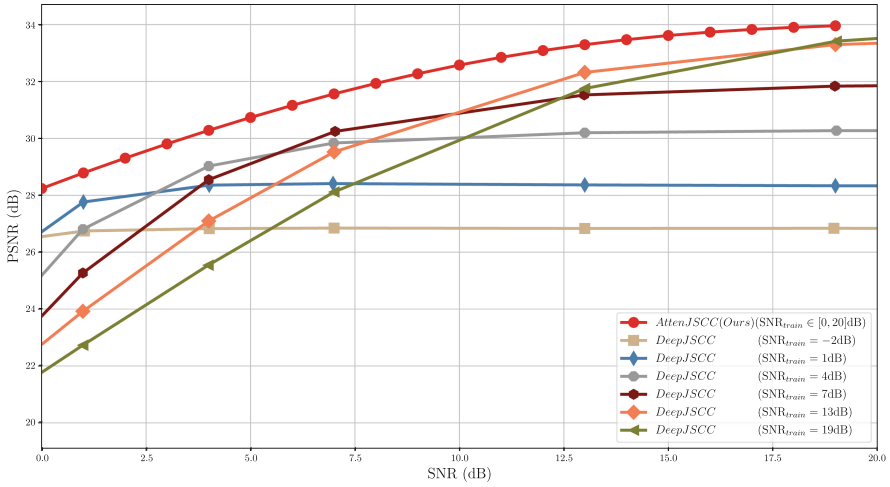
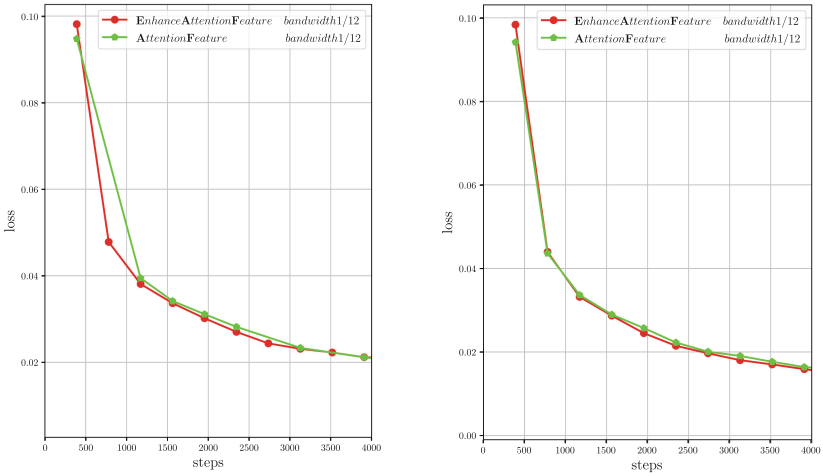


Fig. 6. Performance under different test SNR on Kodak images while R =1/12.

complicated scenarios rather than simple objects which is harder to restore. During evaluation, each image in Kodak dataset is transmitted 30 times to diminish the randomness of channel noise. Figure 6 illustrates that the performance of proposed system is competitive even on larger and complicated images.



(a) Loss curve for R = 1/12

(b) Loss curve for R = 1/6

Fig. 7. Loss curve for different setup. The rest of the network remains same while using different Attention Feature module.

To compare the effectiveness of our proposed enhanced attention feature module, we show the loss curves of two different setups in Fig. 7 which illustrates the loss curve of two different setups. The only difference in the two experiment setup is the attention feature module. We can tell from the loss curve that the proposed module helps to converge to a better solution point faster. Such performance comparison shows that our proposed EAF module captures the inner feature of the image better and thus amplify the weight of core features before interacting with the channel SNR.

5 Conclusion

In this paper, we have proposed a novel deep attention joint source channel coding named AttenJSCC architecture for transmitting images over wireless channels. In this architecture, both the encoder and decoder are designed as complementary convolutional neural networks and are trained jointly to minimize the MSE of the reconstructed images. We implemented relative experiments thus demonstrates validity of proposed EAF module. Also, we evaluated the performance of this AttenJSCC approach against some of the most powerful and famous previous works on both small datasets with single objects and large datasets with complicated scenarios. Our extensive numerical simulations demonstrate that AttenJSCC significantly outperforms particularly in environments with low SNR condition.

Acknowledgment. This research was funded by National Key R&D Program of China (2022YFC3801100) and Heilongjiang Province Key R&D Program (2022ZX01A31).

References

1. Said, D.: A survey on information communication technologies in modern demand-side management for smart grids: Challenges, solutions, and opportunities. IEEE Eng. Manage. Rev. **51**(1), 76–107 (2023)
2. Zhou, R., Tian, C., Liu, T.: Exactly tight information-theoretic generalization error bound for the quadratic Gaussian problem. IEEE J. Selected Areas Inform. Theory **5**, 94–104 (2024). https://doi.org/10.1109/JSAIT.2024.3380598
3. Yang, M., Bian, C., Kim, H.S.: Deep joint source channel coding for wireless image transmission with ofdm. In: ICC 2021 - IEEE International Conference on Communications, pp. 1–6 (2021)
4. Kurka, D.B., Gündüz, D.: Bandwidth-agile image transmission with deep joint source-channel coding. IEEE Trans. Wireless Commun. **20**(12), 8081–8095 (2021)
5. Xuan, Z., Narayanan, K.: Analog joint source-channel coding for gaussian sources over awgn channels with deep learning. In: 2020 International Conference on Signal Processing and Communications (SPCOM), pp. 1–5. IEEE (2020)
6. Bourtsoulatze, E., Burth Kurka, D., Gunduz, D.: Deep joint source-channel coding for wireless image transmission. IEEE Trans. Cogn. Commun. Network. **5**(3), 567–579 (2019). https://doi.org/10.1109/TCCN.2019.2919300
7. Kurka, D.B., Gündüz, D.: Deepjscc-f: Deep joint source-channel coding of images with feedback. IEEE J. Select. Areas Inform. Theory **1**(1), 178–193 (2020)
8. Tung, T.Y., Kurka, D.B., Jankowski, M., Gündüz, D.: Deepjscc-q: Channel input constrained deep joint source-channel coding. In: ICC 2022-IEEE International Conference on Communications, pp. 3880–3885. IEEE (2022)
9. Dai, J., et al.: Nonlinear transform source-channel coding for semantic communications. IEEE J. Sel. Areas Commun. **40**(8), 2300–2316 (2022)
10. Ballé, J., Laparra, V., Simoncelli, E.P.: End-to-end optimized image compression. In: International Conference on Learning Representations (2016)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016)
12. Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., Hu, Q.: Eca-net: efficient channel attention for deep convolutional neural networks. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11531–11539 (2020)
13. Jialong, X., Ai, B., Chen, W., Yang, A., Sun, P., Rodrigues, M.: Wireless image transmission using deep source channel coding with attention modules. IEEE Trans. Circuits Syst. Video Technol. **32**(4), 2315–2328 (2021)
14. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)
15. Crnjanski, J., Krstić, M., Totović, A., Pleros, N., Gvozdić, D.: Adaptive sigmoid-like and prelu activation functions for all-optical perceptron. Opt. Lett. **46**(9), 2003–2006 (2021)
16. Kingma, D.P., Ba, J.: Adam: Amethod for stochastic optimization
17. Sheikh, H.R., Sabir, M.F., Bovik, A.C.: A statistical evaluation of recent full reference image quality assessment algorithms. IEEE Trans. Image Process. **15**(11), 3440–3451 (2006)

18. Kiku, D., Monno, Y., Tanaka, M., Okutomi, M.: Residual interpolation for color image demosaicking. In: 2013 IEEE International Conference on Image Processing, pp. 2304–2308. IEEE (2013)
19. Lin, T.Y., et al.: Microsoft coco: common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pp. 740–755. Springer (2014). https://doi.org/10.1007/978-3-319-10602-1_48