Outdoor Experiment of Deep Joint Source-Channel Coding Using FFT-Enabled Convolutional Neural Network for Image Transmission

Tomoka Mori*, Hiroshi Tatsukawa†, Yuji Kawai†, Yoshinori Shinohara†, Hiroki Ikeda† and Daisuke Hisano*
* Graduate School of Engineering, The University of Osaka, Japan
E-mail: {mori23@pn., hisano@}comm.eng.osaka-u.ac.jp
Tel: +81-6-6879-7728
† Magna Wireless Corporation, Japan
E-mail: {hiroshi.tatsukawa, kawai.yuji, yoshinori.shinohara, hiroki.ikeda}@magna-wireless.co.jp

***Abstract*—Deep Joint Source-Channel Coding (DeepJSCC) has gained attention as a form of semantic communication that conveys not only information but also meaning and intent. It leverages deep learning, utilizing an autoencoder to map information sources such as images directly to IQ symbols. DeepJSCC has been extensively studied for image transmission, demonstrating advantages such as avoiding the cliff effect and achieving a high Peak Signal-to-Noise Ratio (PSNR) even in low Signal-to-Noise Ratio (SNR) regions. Traditionally, convolutional neural network (CNN)-based autoencoders are used in DeepJSCC for image transmission. However, depending on the terminal used, these methods can be computationally intensive. To address this challenge, the authors have proposed FFT-DeepJSCC, which replaces the 2D convolutional layer within the CNN with Fast Fourier Transform (FFT) and element-wise product operations. This modification aims to reduce the computational burden while maintaining performance. The study evaluates the computational load reduction achieved by FFT-DeepJSCC and examines the impact of the altered layer structure on PSNR. Furthermore, the authors validate the method's effectiveness through actual image transmission experiments using a modified commercially available 5G base station and user terminal. The results demonstrate the performance and feasibility of the proposed approach.**

I. Introduction

Deep learning-based joint source-channel coding (DeepJSCC) has been attracting attention in recent years as a highly efficient image transmission technology. DeepJSCC is known to provide higher-quality image transmission than telecommunication systems that combine source coding, such as JPEG2000 and better portable graphics (BPG), with near-Shannon-limit channel coding, such as low-density parity-check (LDPC) coding. In addition, DeepJSCC avoids the cliff effect, which images collapse when the signal-to-noise power ratio (SNR) falls below a certain level. That is, DeepJSCC performs well even at low SNR region. Considering the increase in demand for transmission of high-definition images such as 4K/8K, real-time transmission of high frame rate video, and video/image transmission using small devices without large memory GPUs, a more lightweight DeepJSCC is desired in the future. To meet these requirements, a method using a convolutional neural network (CNN) with about shallow layers is a strong candidate. The study of DeepJSCC, which is lightweight and can operate at high throughput, is expected to accelerate in the future.

We have proposed employing 2D fast Fourier transform (FFT) and element-wise product operations on CNN-based DeepJSCC. The proposed technique reduces the number of kernels in CNN. In this paper, to evaluate the effectiveness of the proposed method, we report the results of outdoor experiments in line-of-sight (LoS) and Non-LoS (NLoS) environments.

II. Related Work

There are a number of previous studies on CNNs with FFT: *Michael Mathieu, et al.* proposed the introduction of FFT in 2014 to speed up CNNs in the training process [12]; *Harry Pratt et al.* in 2017 [13]. Furthermore, in terms of integration into edge devices, *Tahmid Abtahi et al.* show that applying FFT can reduce run-time [14]. On the other hand, signal processing in the frequency domain using FFT usually performs well when the input signal size and kernel size are comparable. In structures such as ResNet used in deep learning, the kernel size is typically 3×3 and the case cannot benefit from the signal processing in the frequency domain. For this reason, *Andrew Lavin* and *Scott Gray* proposed the application of Winograd's minimal filtering algorithms as a speed-up method [15]. However, unlike models with massive multiple layers, such as ResNet, DeepJSCC uses a relatively shallow layered structure. For this reason, the kernel sizes of DeepJSCC are typically larger than 3×3. In other words, a FFT-based CNN may be more sufficient. Against this background, this paper considers FFT-based DeepJSCC as a first step in speed-up.

III. System Model

This section describes the system model of a CNN-based DeepJSCC. Figure 1 shows the layer structure of DeepJSCC. An encoder consists of a two-dimensional convolutional layer (Conv), batch normalization (BN), and activation function, while a decoder includes a two-dimensional transpose convolutional layer (TransConv) instead of Conv. Subscript $H \times W \times F|S$ describing above each layer is a parameter

979-8-3315-7206-8/25/$31.00

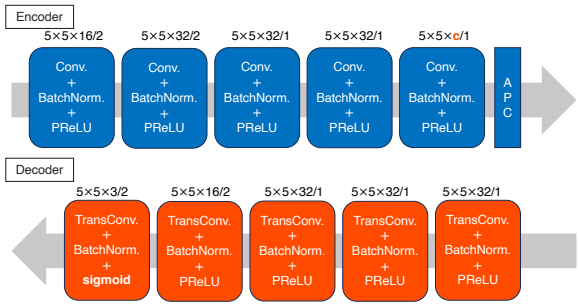


Fig. 1. DeepJSCC layer structure.

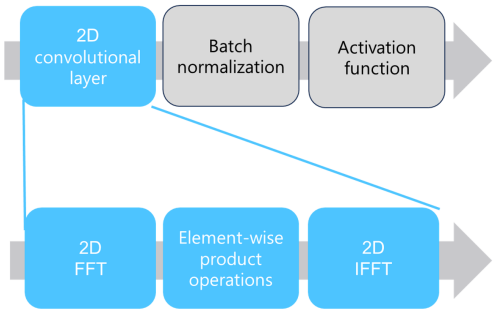


Fig. 2. Proposed FFT-based DeepJSCC method.

of Conv/TransConv where H and W are kernel size, *height* $\times$ *width*. F is the number of outputs. S is stride size. DeepJSCC encoder converts an input image $\mathbf{x}_0 \in P^{h_{\text{in}} \times w_{\text{in}} \times 3}$, $P \in \{\forall i \in \mathbb{Z} | 0 \leq i \leq 255\}$ into a complex baseband signal $\mathbf{z} \in \mathbb{C}^{N_{\text{sym}}}$ That is, the operation is only a simple convolution, the output $\mathbf{x}_i$ at i-th layer is expressed as,

$$\mathbf{x}_i^{(m)} = f_i\left(\sum_{n=1}^{k_i}\left(\mathbf{w}_i^{(m,n)} * \mathbf{x}_{i-1}^{(n)} + \mathbf{b}_i^{(n)}\right)\right), \tag{1}$$

where $\mathbf{w}_i^{(m,n)} \in \mathbb{R}^{h_k \times w_k}$ is a kernel matrix, (m, n) represents the kernel corresponding to the (m)-th output channel for the (n)-th input channel. (m) in $\mathbf{x}_i^{(m)} \in \mathbb{R}^{h_i \times w_i}$ represents the third dimension element of $\mathbf{x}_i$. $(*)$ is a convolutional product. Note that, for simplicity, strides are omitted. $f_i(\cdot)$ is an activation function. The encoder employs average power constraint (APC) on the output from the last layer $\mathbf{x}$ and conducts the flatten operation to $\mathbf{x}$. Finally, we obtain the output IQ symbol as,

$$\mathbf{z} = (\mathbf{U} + i\mathbf{L})\hat{\mathbf{x}} \tag{2}$$

where $\hat{\mathbf{x}} \in \mathbb{R}^{2N_{\text{sym}} \times 1}$ is a flattened output vector. $\mathbf{U}$ and $\mathbf{L}$ are expressed as,

$$\mathbf{U} = [\mathbf{I}\ \mathbf{O}],\ \mathbf{L} = [\mathbf{O}\ \mathbf{I}],$$

where $\mathbf{I}$ is identity matrix, $N_{\text{sym}} \times N_{\text{sym}}$, and $\mathbf{O}$ is zero matrix, $N_{\text{sym}} \times N_{\text{sym}}$. That is, the matrix size of $\mathbf{U}$ and $\mathbf{L}$ are $N_{\text{sym}} \times 2N_{\text{sym}}$. N_{sym} is expressed as,

$$N_{\text{sym}} = \frac{h_{\text{in}} w_{\text{in}} c}{2\prod {s_i}^2}, \tag{3}$$

where c is the number of output images. s_i is the stride size at i-th Conv. The output from the last Conv layer is bisected into I and Q channels. Additive white Gaussian noise (AWGN) is added to $\mathbf{z}$ and the signal is transmitted to the decoder. At the decoder, the image can be restored by the reverse process of the encoder. However, upsampling is conducted by adjusting the stride in TransConv.

IV. Proposed FFT-based DeepJSCC

The proposed scheme replaces the convolutional layer with two-dimensional FFT (2D-FFT), 2D-IFFT, and element-wise product (Hadamard product) as shown in Fig. 2. The matrix input into i-th 2D-FFT layer is assumed to $\mathbf{x}_i \in \mathbb{R}^{h_i \times w_i \times k_i}$ where k_i is the number of channels. For example, when the matrix is RGB image, then $k_i = 3$. Here, to operate in the frequency domain, we must conduct FFT to both the input matrix and the kernel matrix. Since the sizes of the input matrix and the kernel matrix are generally different, we apply zero-padding to the kernel matrix and then FFT is performed. Let the zero-padding matrix be,

$$\mathbf{Z}_{\text{h}} = [\mathbf{O}_{\text{hu}}\ \mathbf{I}\ \mathbf{O}_{\text{hl}}]^\top,$$

$$\mathbf{Z}_{\text{w}} = [\mathbf{O}_{\text{wl}}\ \mathbf{I}\ \mathbf{O}_{\text{wr}}],$$

where $(\cdot)^\top$ stands for transpose, $\mathbf{I} \in \mathbb{R}^{h_i \times w_i \times k_i}$, and the sizes of zero matrix $\mathbf{O}_{\text{hu}}$, $\mathbf{O}_{\text{hl}}$, $\mathbf{O}_{\text{wl}}$ and $\mathbf{O}_{\text{wr}}$ are $h_i \times \lfloor \frac{h_i - h_k}{2} \rfloor$, $h_i \times \lceil \frac{h_i - h_k}{2} \rceil$, $w_i \times \lfloor \frac{w_i - w_k}{2} \rfloor$, and $w_i \times \lceil \frac{w_i - w_k}{2} \rceil$, respectively. h_k and w_k is the first and second dimensions of kernel. The FFT-enabled kernel $\mathbf{w}_i^{(m,n)}$ is represented as $\mathbf{Z}_{\text{h}}\mathbf{w}_i^{(m,n)}\mathbf{Z}_{\text{w}}$.

Next, we conduct FFT to both the input matrix $\mathbf{x}_{i-1}$ and the zero-padded kernel matrix $\mathbf{Z}_{\text{h}}\mathbf{w}_i^{(m,n)}\mathbf{Z}_{\text{w}}$. Using FFT matrix $\mathbf{D}$, the output matrix from i-th layer is expressed as,

$$\mathbf{x}_i^{(m)} = f_i\left(\mathbf{D}^{-1}\sum_{n=1}^{k_i}\left\{\mathbf{D}\mathbf{Z}_{\text{p}}\mathbf{w}_i^{(m,n)}\mathbf{Z}_{\text{p}}^\top \odot \mathbf{D}\mathbf{x}_{i-1}^{(n)} + \mathbf{b}_i^{(n)}\right\}\right), \tag{4}$$

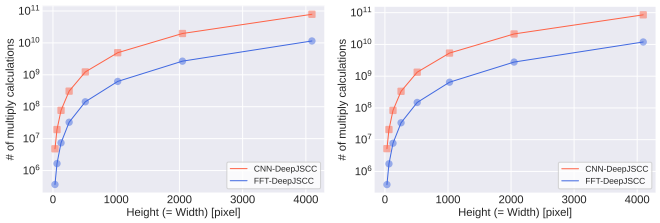
where $\odot$ is Hadamard product. For m and n, the same as in the previous section. $\mathbf{b}^{(n)}$ is the bias matrix.

To evaluate the effectiveness of the proposed scheme, we firstly estimated the amount of multiplications that can be reduced and investigated the impact on image quality based on peak signal-to-noise power ratio (PSNR).

V. Numerical Analysis

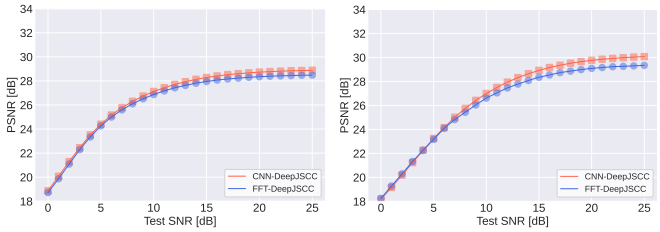
A. Estimation of Complexity

We estimate the complexity reduction of the proposed FFT-DeepJSCC. The Conv layer is replaced with FFT, Hadamard product, and IFFT. We assume height, width, and the number of channels of the input data at i-th layer to h_{i-1}, w_{i-1}, k_{i-1}, respectively. Those of the output data are assumed to h_i, w_i, k_i. Firstly, we estimate the number of multiplications of the conventional CNN-based DeepJSCC when the CNN is



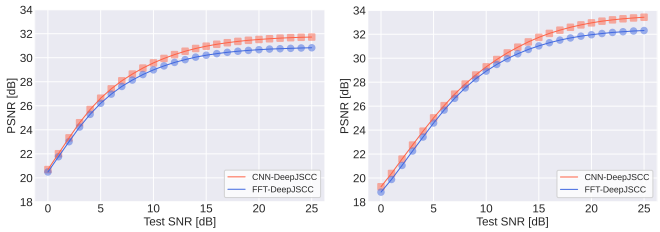
(a) $c = 8$. (b) $c = 16$.

Fig. 3. Comparison of multiplications.



(a) $\gamma_{\text{train}} = 8$ dB. (b) $\gamma_{\text{train}} = 13$ dB.

Fig. 4. SNR vs. PSNR at $c = 8$.



(a) $\gamma_{\text{train}} = 8$ dB. (b) $\gamma_{\text{train}} = 13$ dB.

Fig. 5. SNR vs. PSNR at $c = 16$.

implemented naively. Assuming kernel size to h_i', w_i', $k_{i-1}k_i$, the number of multiplications N_{CNN} is expressed as,

$$N_{\text{CNN}}^{(i)} = \frac{h_{i-1}w_{i-1}h_i'w_i'k_{i-1}k_i}{s_{h,i}s_{w,i}}, \tag{5}$$

where $s_{h,i}$ and $s_{w,i}$ are stride size at the direction of height and width. Note that we assume the padding operation to keep the size of height and width.

Next, we estimate the number of multiplications of the proposed FFT-based DeepJSCC. The number of multiplications is the sum of the FFT, the Hadamard product, and the IFFT, and is expressed as,

$$\begin{aligned} N_{\text{FFT}}^{(i)} = & \underbrace{k_{i-1}h_{i-1}w_{i-1}\log_2 h_{i-1}w_{i-1}}_{\text{FFT}} \\ & + \underbrace{h_{i-1}w_{i-1}k_{i-1}k_i}_{\text{Hadamard}} \\ & + \underbrace{k_i \frac{h_{i-1}w_{i-1}}{s_{h,i}s_{w,i}} \log_2 \frac{h_{i-1}w_{i-1}}{s_{h,i}s_{w,i}}}_{\text{IFFT}}. \end{aligned} \tag{6}$$

The output data from Hadamard product is cropped at the direction of height and width. This operation is equivalent to the stride operation.

Figure 3 shows the estimated number of multiplications for the proposed scheme (FFT-DeepJSCC) and the conventional scheme (CNN-DeepJSCC). The horizontal axis is the height of the input image. Since this calculation assumed a square image, the width of the image has the same number of pixels as its height. The FFT-DeepJSCC can reduce the number of multiplications by about one order of magnitude. Meanwhile, this did not take into account parallel processing such as GPU, so the actual computational complexity of the CNN is expected to be less than this result. However, sufficient parallel processing performance may not be achieved if only CPUs are used for low cost and low power consumption, or if devices with small memory and running on GPUs with low clock frequency are used. This evaluation is especially useful enough for use in IoT devices under such constraints.

B. PSNR Evaluation

PSNR γ_{psnr} is introduced as the image quality metric and is defined as

$$\gamma_{\text{psnr}} = 10\log_{10}\frac{R_{\max}^2}{e_{\text{mse}}}, \tag{7}$$

where $R_{\max}$ is the maximum pixel value. In the common case, $R_{\max} = 255$. e_{mse} is the MSE between the input and output images. It is defined as,

$$e_{\text{mse}} = \frac{1}{hw}\sum_{i=1}^{h}\sum_{j=1}^{w}[x(i,j) - x'(i,j)]^2. \tag{8}$$

Figure 4 and Figure 5 show the simulation results. The horizontal and vertical axes are the signal-to-noise ratio (SNR) in the test phase and PSNR, respectively. The comparison between CNN-DeepJSCC and the proposed FFT-DeepJSCC shows that there is a slight degradation of the performance of FFT-DeepJSCC, which is less than 1.0 dB. The main reason for this degradation is the slight difference in kernel size between CNN-DeepJSCC and FFT-DeepJSCC. Specifically, it is considered that the kernel size of FFT-DeepJSCC is slightly different from that of CNN-DeepJSCC because FFT-DeepJSCC switches to processing in the frequency domain. Furthermore, in the downsampling process, average pooling was applied in FFT-DeepJSCC, but not in CNN-DeepJSCC, and we expect this difference affects the performance. Specifically, it is expected that some information is lost by performing average pooling, resulting in a slight degradation of the performance.

VI. Experiments

A. Experimental Setup

We conducted an empirical experiment to compare the image quality of the proposed method, FFT-DeepJSCC, with that of CNN-DeepJSCC. In this study, c, which indicates the number of output channels at the end of the encoder, is set to 8, and an additive white Gaussian noise (AWGN) communication channel is assumed during the learning process, and two different SNR conditions, 8 dB and 13 dB, are used for the

TABLE I
LEARNING PARAMETERS OF DEEPJSCC.

Item	Value
Epochs	10000
batch size	1000
Training SNR	8,13 dB
Pooling	Average
Optimizer	Adamax
Loss function	Mean Squared Error
Metrics	PSNR

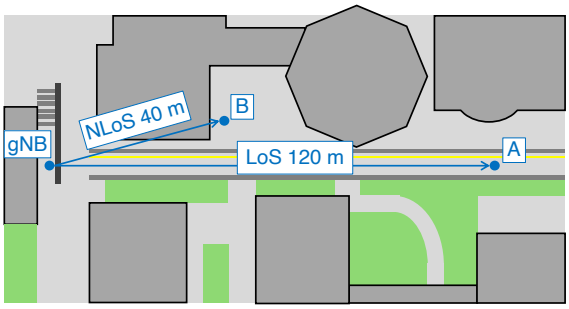


Fig. 6. Experimental condition.

(a) Transmission side. (b) Reception side.

Fig. 7. Experiment.

verification. In this learning process, parameters such as the number of epochs, patch size, and PSNR, an evaluation index, are shown in Table I.

The base station (gNB) and user terminal (UE) used in the experiment were private 5G gNB and UE commercially available in Japan released by Magna Wireless corp. These devices were modified to be able to transmit and receive DeepJSCC signal [11], [16]. On the receiver side, channel estimation was performed using the pilot signal in the common process in 5G, and the IQ signal was extracted after frequency domain equalization. Figure 6 shows the experimental overview. Figure 7 shows photographs of the installation of the transmitter and receiver devices during the data acquisition. We conducted the experiments at three points A and B in Suita campus of the university of Osaka, Japan. The point A obtained the line of sight (LoS) environment. On the other hand, the points B formed non-LoS (NLoS) environment. These conditions were recorded in detail, including the measurement of RSSI values, which represent the reception strength of radio signals, and the

TABLE II
EXPERIMENTAL CONDITIONS

Point	Distance [m]	RSSI [dBm]
A (LoS)	120 m	−65 dBm
A (LoS)	120 m	−72 dBm
A (LoS)	120 m	−75 dBm
B (NLoS)	40 m	−80 dBm

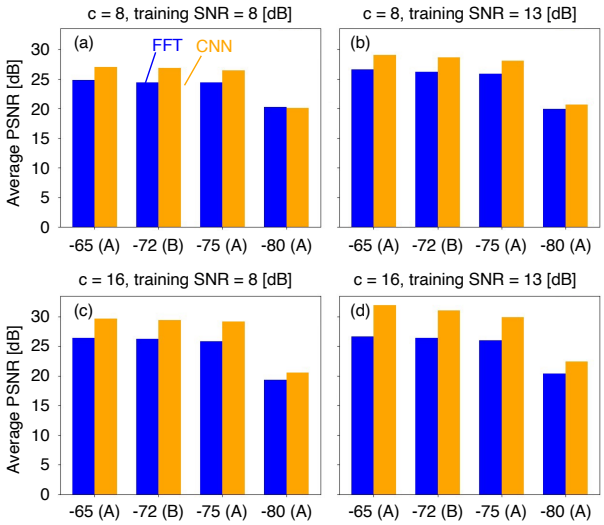


Fig. 8. Experimental results

results are summarized in Table II.

B. Results Experimental Evaluation

Figure 8 shows the experimental results. The figure plots PSNR on the vertical axis and four experimental conditions on the horizontal axis. The four conditions include data for comparing the obtained PSNR values for both the proposed FFT-DeepJSCC and the conventional CNN-DeepJSCC. FFT-DeepJSCC shows a degradation of characteristics in the range from 1.0 dB to 3.0 dB compared to CNN-DeepJSCC. These results obtain the same trend as well as the simulation ones. Even when the proposed FFT-based scheme is used, the performance is sufficient to withstand the actual environment.

VII. CONCLUSION

We have proposed a DeepJSCC method utilizing FFT and element-wise product operations, demonstrating its effectiveness in reducing the number of multiplications. The proposed method significantly reduces computational complexity compared to conventional convolution-based approaches. Meanwhile, a slight degradation in image quality was observed less than 1.0 dB in simulations. This paper conducted the outdoor experiment using private 5G base station and user terminal. The experimental results indicated the PSNR degradation between 1.0 to 3.0 dB. Future work will focus on implementing the proposed method on the decoder side to enhance overall performance. Improvements in decoder processing are expected to increase image quality and further validate the method's effectiveness. Specifically, we plan to

refine the layer structure to achieve more accurate restoration. Additionally, we will evaluate the execution speed of the proposed method on GPUs and FPGAs, assess its processing efficiency in real-world operational environments, and identify potential challenges for practical application.

ACKNOWLEDGMENTS

A part of this paper is based on results obtained from "Research and Development Project of the Enhanced Infrastructures for Post-5G Information and Communication Systems" (JPNP20017), commissioned by the New Energy and Industrial Technology Development Organization (NEDO). A part of this work was supported by JST, ACT-X Grant Number JPMJAX24MA, Japan and by The Telecommunications Advancement Foundation (TAF), Japan.

REFERENCES

[1] E. Bourtsoulatze, D. B. Kurka, and D. Gündüz, "Deep Joint Source-Channel Coding for Wireless Image Transmission," *IEEE Trans. on Cog. Commun. Netw.*, vol. 5, no. 3, pp. 567–579, 2019.

[2] D. B. Kurka and D. Gündüz, "DeepJscc-f: Deep Joint Source-Channel Coding of Images with Feedback," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 178–193, 2020.

[3] NTT docomo, "White Paper 5G Evolution and 6G," pp.5-6, Nov. 2022.

[4] Y. Shao and D. Gündüz, "Semantic Communications with Discrete-Time Analog Transmission: A PAPR Perspective," *IEEE Wirel. Commun. Lett.*, vol. 12, no. 3, pp. 510–514, 2022.

[5] H. Wu, Y. Shao, K. Mikolajczyk, and D. Gündüz, "Channel-Adaptive Wireless Image Transmission with OFDM," *IEEE Wirel. Commun. Lett.*, vol. 11, no. 11, pp. 2400–2404, 2022.

[6] H. Hu, X. Zhu, F. Zhou, W. Wu, R. Q. Hu, and H. Zhu, "One-to-Many Semantic Communication Systems: Design, Implementation, Performance Evaluation," in Proc. of *IEEE Wirel. Commun. Lett.*, vol. 26, no. 12, pp. 2959–2963, 2022.

[7] S. Inokuma, Y. Sasaki, D. Hisano, Y. Nakayama, and K. Maruta, "Performance Evaluation of MIMO Transmission in Deep Joint Source-Channel Coding," *IEEE 99th Vehicular Technology Conference (VTC-Spring)*, 2024.

[8] T. -Y. Tung and D. Gündüz, "DeepWiVe: Deep-learning-aided wireless video transmission," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 9, pp. 2570–2583, 2022.

[9] S. Ibuki, T. Okamoto, T. Fujihashi, T. Koike-Akino, Toshiaki and T. Watanabe, "Rateless Deep Graph Joint Source Channel Coding for Holographic-Type Communication," in Proc. of *IEEE Global Communications Conference (GLOBECOM)*, pp. 3330–3335, 2023.

[10] M. Liu, W. Chen, J. Xu, and B. Ai, "Real-Time Implementation and Evaluation of SDR-based Deep Joint Source-Channel Coding," in Proc. of *IEEE 96th vehicular technology conference (VTC2022-Fall)*, pp. 1–5, 2022.

[11] K. Matsumoto, Y. Inoue, Y. Hara-Azumi, K. Maruta, Y. Nakayama, Y. Shinohara, H. Ikeda, D. Hisano, "Implementation of Deep Joint Source-Channel Coding on 5G Systems for Image Transmission," in Proc. of IEEE *98th Vehicular Technology Conference (VTC2023-Fall)*, 2023.

[12] M. Mathieu, M. Henaff, and Y. LeCun, "Fast Training of Convolutional Networks through FFTs," in Proc. of *International Conference on Learning Representations (ICLR2014)*, 2014.

[13] H. Pratt, B. Williams, F. Coenen, and Y. Zheng, "FCNN: Fourier Convolutional Neural Networks," in Proc. of *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD*, pp 786–798, 2017.

[14] T. Abtahi, C. Shea, A. Kulkarni, and T. Mohsenin, "Accelerating Convolutional Neural Network with FFT on Embedded Hardware," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 9, pp. 1737–1749, 2018.

[15] A. Lavin and S. Gray, "Fast Algorithms for Convolutional Neural Networks," in Proc. of *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4013–4021, 2016.

[16] D. Hisano, K. Matsumoto, Y. Inoue, Y. Hara, K. Maruta, Y. Nakayama, H. Tatsukawa, Y. Kawai, Y. Shinohara, and H. Ikeda, "5G Indoor/Outdoor Field Trial of Deep Joint Source-Channel Coding Method," in IEEE Open Journal of the Communications Society, doi: 10.1109/OJCOMS.2025.3576502.