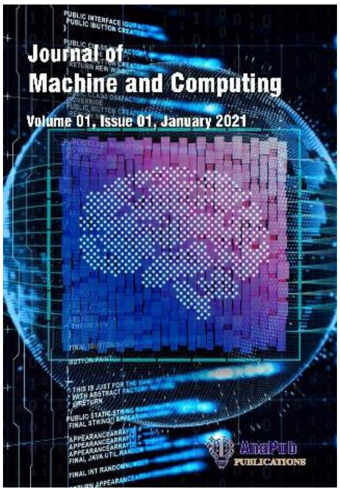
Journal Pre-proof

FPGA-Based Image Compression for Wireless Communication Networks using - CRAN Architecture



Lakshmisha S K, Madhusudhan M V, Goutami Chenumalla, Impa B H, Bhavana A and Laxmi Singh

DOI: 10.53759/7669/jmc202505176

Reference: JMC202505176

Journal: Journal of Machine and Computing.

Received 30 March 2025

Revised from 02 May 2025

Accepted 29 July 2025

FPGA-Based Image Compression for Wireless Communication Networks using - CRAN Architecture

LAKSHMISHA S K[1], MADHUSUDHAN M V[2], GOUTAMI CHENUMALLA[3], IMPA B H[4] BHAVANA A[5], LAXMI SINGH[6]

[1, 2, 4, 5] Department of CSE, Presidency University, Bengaluru, India

[3] Department of CSE, BMSIT&M Bengaluru, India

[6]Department of ECE, Rabindranath Tagore University, Bhopal India

E-mail: [1] lakshmisha.sk@presidencyuniversity.in, [2] mv.madhu@gmail.com
[3] goutamich@bmsit.in [4] impabh.18@gmail.com, [5] bhavana.a.research@gmail.com,
[6] laxmi15singh@gmail.com

ABSTRACT

This work introduces an Field Programmable Gate Array (FPGA) based image compression method utilizing Huffman coding (FICH) to enhance the efficiency of wireless networks, particularly within the Cloud-based Radio-Access-Network (C-RAN) architecture. The FICH method addresses image compression challenges in C-RAN, offering faster compression and decompression times compared to existing FPGA approaches. The findings include significant improvements in Bit-Error-Rate (BER), Symbol-Error-Rate (SER), and Error-Vector Magnitude (EVM), with average BER, SER, and EVM improvements of 37.85%, 24.64%, and 24.56% for fewer RRHs, and 96.10%, 91.1[illegible]%, and [illegible]8.72% for more RRHs, respectively. Additionally, the FICH method demonstrated reduced encoding and decoding times, averaging 0.0545 seconds versus 0.0853 seconds when compared with existing approach. The approach also ensures robust and scalable compression, optimizing resource utilization with FPGA-based hardware acceleration. These advancements support the growing data demands of modern wireless networks.

Keywords: Encoding, Decoding, Image Compression, FPGA, Huffman coding, Radio Access Network

1. INTRODUCTION

The growing popularity of technological advancement in wireless technology and online streaming of videos has prompted the development of more compact wireless networks capable of handling the ever-increasing data rates. Hence, because of this, there has been increased Inter-Cell-Interference (ICI) due to the decreasing range among Remote-Radio-Units (RRUs) [1]. One promising structure for dealing with the dominating ICI includes the Cloud-based Radio-Access-Network (C-RAN), which allows the Central-Processing-Unit (CPU) to execute joint pre-coding during downlink transmission along with joint de-coding during uplink reception [2]. Through the use of virtualization, C-RAN design is able to manage an unlimited number of RRUs within the network's infrastructure. Figure 1 shows the components of a C-RAN framework, which includes RRU networks, a pool of Base-Band-Units (BBUs) including a transportation network known as fronthaul. Every RRU has a fixed allocation from the shared BBU pool. The BBU is responsible for managing the computational resources of RRU networks, which links different wireless devices, and it has powerful storage and computing abilities.

The C-RAN is a novel and innovative design for 5G wireless networks and mobile networks. It can meet a lot of needs, including lowering system costs, improving the utilization of energy, increasing throughput, and decreasing latency [3]. In comparison to the costly and time-consuming Micro Base-Station (MBS), the cost, area, and effort required for setting up RRUs using C-RAN is far lower. In addition, it enables MBSs along with users to save energy by transferring operations that use a lot of power to a neighboring cloud [4]. In addition, RRUs associated with the very same cloud can accomplish better spectral effectiveness and have easier implementation of synchronized multi-point transmissions [5]. On top of that, C-RAN platforms can minimize the latency that comes from doing different kinds of processes. For instance, handovers can be executed more quickly within the cloud than among Base-Stations (BS) using Open-RAN (O-RAN) [6].

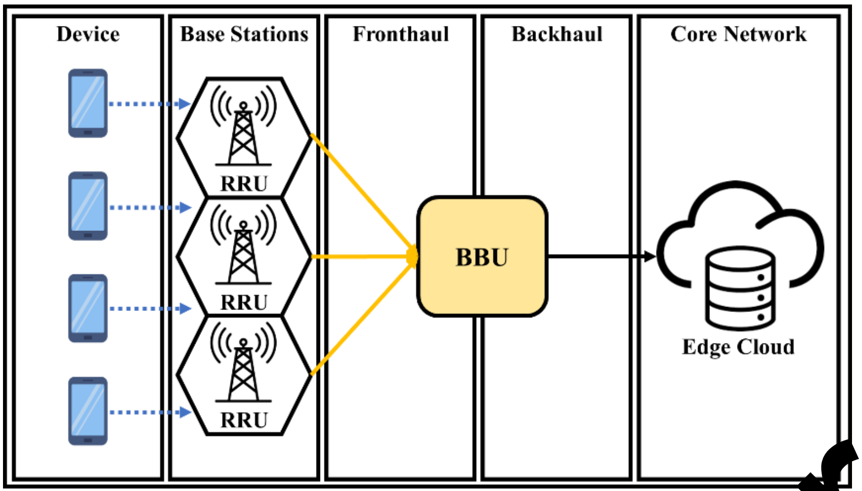


Figure 1. Architecture of Cloud-based Radio-Access-Network (C-RAN).

Communication between the RRUs and BBUs can take place through cellular, millimeter-wave, or optical fiber connections, and using other approaches also. Moreover, the optical fiber connection allows for the low-cost deployment of cellular and millimeter communications. Nevertheless, they result in increased latency along with reduced bandwidth [7]. Furthermore, the links between BBU and RRU is called the fronthaul. In order to set up C-RAN, the most important consideration is the ability needs of the fronthaul link. Among the most extensively researched methods in uplink C-RAN, for instance, is Quantization-and-Forward (QF) [8]. Before sending the signal with quantization to a CPU over the limited capability fronthaul, every RRU processes and quantifies it. Finding the best compression strategy using limited resources while minimizing distortion-error is difficult. Also, if there are a lot of observations, then the task will become even more challenging [9]. This issue was formalized in [10] using only one RRU and Wyner-Ziv compressing approach with specific assumptions, for instance utilizing Gaussian channels or signals. But this compression strategy is difficult to apply in real-world networking systems because of the high computational costs and latency caused by an indefinite block size coding.

Recent years have seen widespread usage of Deep-Learning (DL) and Artificial-Intelligence (AI) to address a wide range of practical issues. Some of the increasingly cutting-edge DL algorithms utilized for handling recognizing problems in various settings is Convolutional-Neural-Networks (CNNs) [11]. Compared to traditional algorithms, CNNs provide better accuracy. On the other hand, a lot of processing and memory capacity is needed for the convolutional process [12]. Given the high-power consumption utilization by CNN, this presents a computational issue to the CPU. On the other hand, hardware acceleration technologies like Application-Specific Integrated-Circuit (ASIC), Graphical-Processing-Unit (GPUs), and Field-Programmable Gate-Array (FPGAs) are being employed to boost CNN throughput [13]. Latency is lowered and energy usage can be reduced when CNNs are implemented using hardware acceleration technologies. GPUs remain among the most popular processors because they enhance CNN inference and training. The problem is that GPUs use too much power, resulting in an important indicator of system efficiency in today's digital devices [14]. Although they are more expensive and take longer to manufacture, ASIC architectures have minimal power consumption along with substantial throughput. But FPGAs boost hardware resource utilization, allowing for hundreds of thousands of floating-point computation processing units with reduced energy consumption. Hence, FPGA-based acceleration devices, similar to ASICs, are a cost-effective and efficient substitute that provides great adaptability and throughput with minimal power usage.

Advancements in FPGA-based hardware acceleration have prompted new developments in methods aimed at enhancing CNN accuracy. More complicated and time-consuming convolution process settings are needed for state-of-the-art CNN algorithms. In order to achieve a balance between efficiency and precision, several object recognition sequence techniques are being designed. These include the You-Only-Look-Once (YOLO) approach [15], Reconfigurable-CNN approaches [16], [17]. Nevertheless, there are limitations when using CNN edge-computing given its complexity and demands of additional operations. There has been a lot of interest in CNN compressing models approaches as a potential solution for these issues. CNN compressing approach streamlines the design of DL approach by decreasing overall parameters, computation, bits and storage requirements needed for inference, and reducing overall complexity. The utilization of edge devices helps in rapid response time, limited memory utilization, and minimal power consumption. Model compression in DL has the potential to streamline network inference, decrease system storage requirements, and mitigate system energy usage. By lowering the cost of edge devices, enhancing effectiveness, and enhancing environmentally friendly sustainability, model compression can boost system competitiveness in the forthcoming wireless network applications scenarios with significant DL approaches [18]. From all the above issues presented, the main aim of this work is as follows

- Understand the current FPGA-based image compression approaches for wireless networks and its application in order to design an approach for fronthaul compression for next-generation wireless networks.
- Present a FPGA-based image compression using Huffman approach (FICH) which consumes less time for image compression for next-generation wireless networks (C-RAN).
- Evaluate and compare the proposed work with existing approaches in terms of Bit-Error-Rate (BER), Symbol Error-Rate (SER) and Error-Vector Magnitude (EVM).

The manuscript is organized in the following way. In Section II, the FPGA-based image compression approaches are discussed in detail. Further, in Section III, a fast FPGA-based image compression approach using Huffman for C-RAN is presented. In Section IV, the fast FPGA-based image compression approach has been evaluated and compared with existing approaches. Finally, in Section V, the conclusion of the work and future work is discussed.

2. LITERATURE SURVEY

In this section, the existing FPGA-based image compression approaches for wireless networks are presented. M. Zhang et al., in [19], presented a compression optimizing approach for CNNs using FPGA. They considered the ImageNet dataset for this study. In this compression approach, initially they reduced the overall parameters of CNN using AlexNet. Further, they utilized two approaches, i.e., quantization and peak-pruning for reducing loss and achieving better compression rate. The evaluation was done on a FPGA board. From results it was seen that the compression approach compressed an image and reduced the size of original image by 3.58%. Y. Barrios et al., in [20], presented a compression approach for hyperspectral image for sending images from space. This work utilized High-Level-Synthesis (HLS) approach for increasing the compression rate. Moreover, this work has presented a reconfigurable-multi-accelerator framework called as ARTICo3 for deployment of their compression approach. For evaluation of their work, they utilized AVRIS where they found that their approach can reach PSNR of an average of 75.[illegible]3%. M. Ledwon et al., in [21], this work focused on lossless-data compression where they utilized FPGA-based accelerators for deflate decompression and compression. For encoding and decoding, this work utilized Huffman, LZ77 and byte packer. Findings show that this work achieved 11% higher throughput during compression in comparison with existing approaches

.

S. Jang, in [22], presented a fast-processing CNN which utilizes the acceleration using FPGAs. The CNN utilizes the parallel and pipeling process from the FPGA for making the compression faster. The work has been compared with different DL approaches like ResNet-50, MobileNetV1 and others. Finding show that the CNN achieved faster compression time in comparison with existing approaches, i.e., achieved accuracy of 68.65%. Also, they tested different DL approaches on FPGA where they found that MobileNetV1 achieved better accuracy for compression, i.e., [illegible]3% of accuracy. K. Pranitha, in [23], presented a compression approach using Discrete-Wavelet Transform (DWT) using FPGA. The DWT utilized an entropy encoding approach for reducing spatial redundancy among the wavelet coefficients and compress using de-correlated data having higher compression. This work utilized Binary Arithmetic-Entropy Coder (BAEC) for designing lossless-compression. Findings show that the approach achieved better frequency and throughput for compression. T. A[illegible] in [24], compared the different approaches for making the process of image compression faster. In this work they considered FPGA, CPU and GPUs for making the process faster. They utilized two DL algorithms, i.e., Scale-Invariat Feature-Transform (SIFT), ResNet50 and MobileNetV2. Findings showed that the GPU and FPGA reduced the time for compression and energy.

Y. [illegible] et al., in [25], presented a neural network called as ResBinESPCN for enahcning the image. Their architecture reduced energy consumption at both software and hardware level. Also, the memeory utilization was reduced. Findings showed that when the ResBinESPCN execution was done on CPU it took more time and utilized more resources. When utilized FPGA, the ResBinESPCN reduced time, reduced resource and energy consumption. Also, they achieved a Peak-Signal to Noise-Ratio (PSNR) of 27.30. M. B. Altman et al., in [26], they have done a survey on Machine-Learning (ML) approaches which utilized FPGAs for implementation. This work mainly surveys on the work related to the healthcare technologies. In this article they came to ac conclusion that by utilizing FPGAs, the image compression, image resolution can be improved.

R. Ghodhbani et al., in [27], presented an approach for compression and decompression of images. This work mainly utilized FPGA for faster process. They utilized the concept of pipeline pause process for resolving the problem of coding-errors. Also, a parallel-block compression approach was proposed for compression and reducing the time. The findings show that it achieves better compression ratio and reduced the frequency of CPU.

H. Sun et al., in [28], main focus was to utilize the learned image-compression approach and FPGA for reducing power and achieving faster image compression. They proposed an algorithm where they used concept of parallelism. They have evaluated their work using Kodak datasets which is open accessible on Kaggle. The results show that the proposed approach was 1.5 times better in comparison with existing approaches. From all the above study it is seen that most of the work utilize the FPGA for hardware acceleration, but very less work has been done for image compression for wireless network. Hence, in this work we present a model for image compression for wireless networks. The model is discussed in detail in the next section.

3. METHODOLOGY

This study is mainly focused on providing FPGA-based image compression using Huffman approach (FICH) for wireless network where the compression of image is faster in comparison with existing approaches. This work utilizes an encoding approach which can effectively detect the errors during the process of encoding in order to boost the image compression process and provide better reconstructed image to the receiver. Architecture diagram of the proposed FICH method is as shown in the Figure 1.

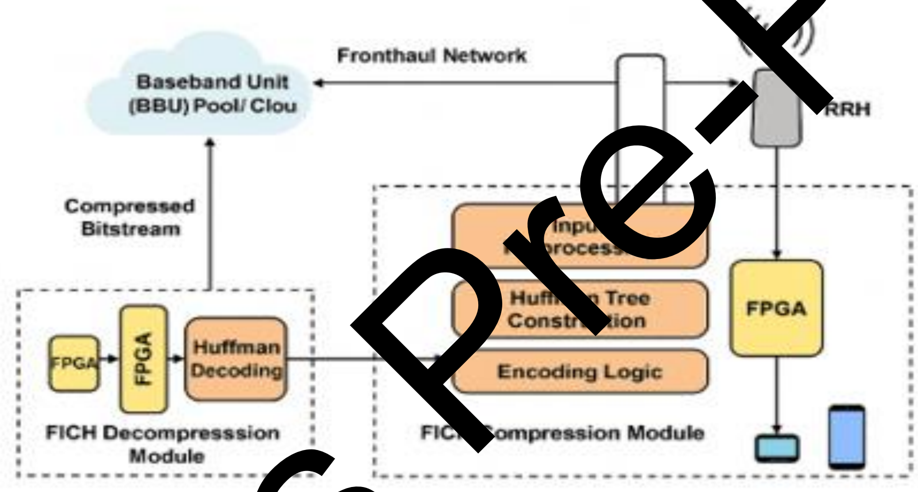


Figure 1. Proposed FICH method architecture diagram

This work utilizes the C-RAN architecture (Figure 1) for efficiently compressing the image using fronthaul compression. In C-RAN, consider that each RRHs has O antenaas which receive images from various users.

Consider an image represented as a_o which goes to the o^{th} antenna, then the image can be represented using the matrices. This is represented using Eq. (1)

$$a_o[p] = \sum_w z_w[p] \times j_{o,w}[p] + y_o[p], \text{ where } o \in 1,2,\ldots,O,\ p \in 0,1,2,\ldots \qquad (1)$$

In Eq. (1), z_w represents frames of Orthogonal-Frequency-Division-Multiplexing (OFDM) as bits received by w^{th} device, $j_{o,w}$ represents the noise which is attained during the transmission from the w^{th} device to o^{th} antenna, y_o represents the gaussian-noise attained when the image is present in o^{th} antenna and $(\times)$ represents the convolution. The overall matrix for the image can be represented using Eq. (2).

$$A = \begin{bmatrix} a_1[0] & a_2[0] & \ldots & a_O[0] \\ a_1[1] & a_2[1] & \ldots & a_O[1] \\ \vdots & \vdots & \ddots & \vdots \\ a_1[P-1] & a_2[O-1] & \ldots & a_O[P-1] \end{bmatrix} \qquad (2)$$

In Eq. (2), P represents the bits of images where the compression needs to be performed. The $a_1, a_2 \ldots, a_k$ matrix is defined using Eq. (3).

$$a_k = [a_k[0] \quad a_k[1] \quad \dots \quad a_k[P-1]]^U \tag{3}$$

In Eq, (3), , $k \in \{1,2,\dots,O\}$ provide better correlation, hence, A is evaluated using low-rank estimation approach. The estimation approach is done using Eq. (4)

$$A = A_0 + G \tag{4}$$

In Eq. (4), it is considered that after the low-rank estimation, $A_0 \in \mathbb{E}^{P*O}$ belongs to the matrix where there exists no noise in the image. Also, the low-rank estimation matrix is considered to have data of image in the form of bits represented as a, behaviour of RRH channel k. In Eq. (4), similar to A_o, G is also considered as $G \in \mathbb{E}^{P*O}$ where it defines the gaussian-noise. Using Eq. (4), the bits of image are compressed and sent towards BBU using the fronthaul. Fig-3.3 for decompression process, the image is decompreesed in the similar way how the compression has been done for achieving A at the BBU.

This study mainly aimed at providing faster image compression process for wireless network (C-RAN); hence it is important to decrease the matrix-size utilizing the low-rank estimation approach. From Eq. (4), it is known that low-rank matrix $A_0 \in \mathbb{E}^{P*O}$, hence it can be said that $P > O > N$. From this hypothesis, the A can be reformulated as Eq. (5) and represented as A''.

$$A'' = \underset{\mathcal{R}ank(\hat{A})=N}{\operatorname{argmin}} \left\|A - \hat{A}\right\|H \tag{5}$$

The Eq. (5) utilizes normailization approach, i.e., Frobenius distance approach ($\|.\|H$). Also, the $\widehat{A}$ is represented using Eq. (6).

$$\widehat{A} = W_N \beta_N X_N^J \tag{6}$$

Where W_N, β_N, and X_N^J are represented using Eq. (7), Eq. (8) and Eq. (9) respectively.

$$W_N = [w_1 \quad w_2 \quad \dots \quad w_N] \tag{7}$$
$$\beta_N = \mathcal{D}iag[\alpha_1 \quad \alpha_2 \quad \dots \quad \alpha_N] \tag{8}$$
$$X_N^J = [x_1 \quad x_2 \quad \dots \quad x_N] \tag{9}$$

The $\widehat{A}$ represents decomposition process [29] where $\widehat{A}$ defines A'' to conjugate transposed matrix $(.)^J$. The Eq. (7), $w_1, w_2, \dots, w_N$ represents eignevectors at leftside and $w_N \in \mathbb{E}^P$. In Eq. (8) $\alpha_1, \alpha_2, \dots, \alpha_N$ represents decomposition values diagonally. In Eq. (9), $x_1, x_2, \dots, x_N$ represents eignevectors at rightside and $x_N \in \mathbb{E}^P$. Using the noise matrix presented in Eq. (4), N can be obtained using [30]. By utilizing the decomposition, the proposed approach obtains N using X_N^J and then it is multiplied to X_N^J so that the transformed matrix A can be obtained. From this operation, the image bits a_k concering O towards the N has no correlation to image. Hence, this operation can be formulated using a matrix as R_N which is presented in Eq. (10).

$$R_N = AX_N = W_N \beta_N \tag{10}$$

In Eq. (10), R_N is considered as $R_N \in \mathbb{E}^{P*N}$ and $N \in \{1,2,\dots,n\}$. The R_N represents the matrix which has no correlation with image. Further, the matrix R_N is compressed by utilizing the Huffman approach [31] and transmitted further towards BBU utilizing the fronthaul. Th A'' is obtained at BBU can be represented using Eq. (11).

$$A'' = R_N X_N^J = W_N \beta_N X_N^J \tag{11}$$

Further, the overall image bits transmitted to fronthaul of C-RAN wireless network is represented using Eq. (11).

$$\mathrm{T} = ON + PN \tag{12}$$

The total compression for the image bits a_o is evaluated using Eq. (12).

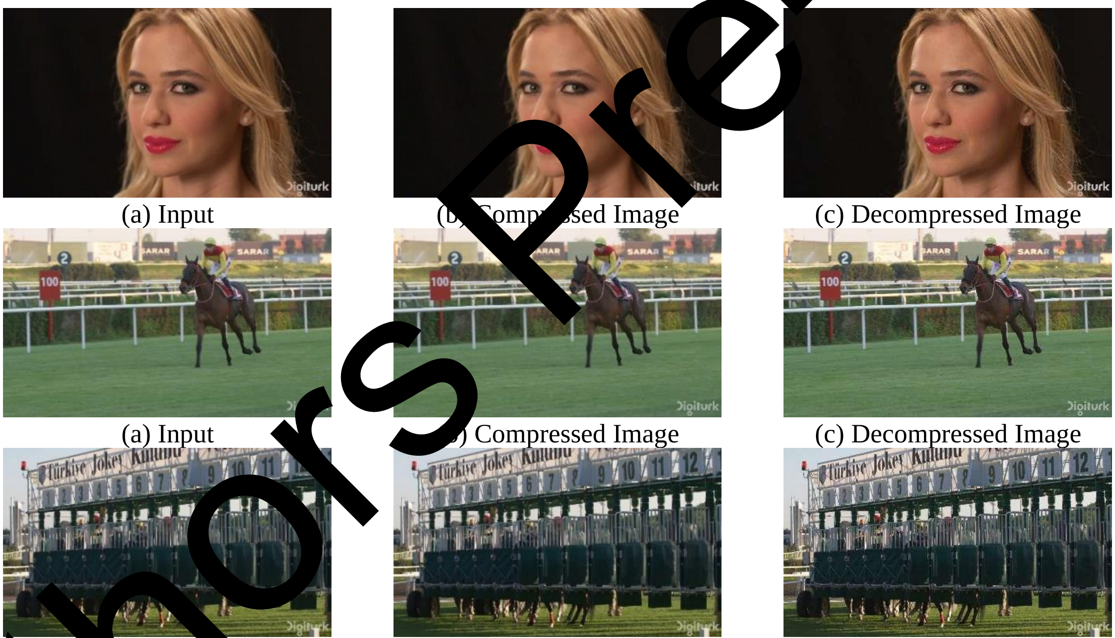
$$\mathcal{CR}_{\mathcal{P}} = \frac{O \times P}{N[O + P]} \quad (13)$$

4. RESULTS AND DISCUSSION

In the proposed FICH architecture, the Compression Time is evaluated by combining both the Encoding and Decoding phases. The performance is assessed using metrics such as Bit Error Rate (BER), Symbol Error Rate (SER), Error Vector Magnitude (EVM), and the number of Remote Radio Heads (RRHs). The evaluation is conducted using datasets of varying sizes, including a few vs. many images and video sequences, with experiments performed on standard datasets like the Kodak dataset and the Ultra Video Group (UVG) dataset [32] [33]. The UVG dataset consists of 16 video sequences having a resolution of 2160p. All the sequences have 600 frames except two having 300 frames.

The duration of each sequence varied from 2.5s to 12s. Also, the frame rate for the sequences varied from 120fps (frame-per-second) to 50fps. The contrast of the images were mixed, i.e., high and low. For all the sequences, the bit-depth was set at 8 and 10. Some sequences had complex structure and some had smooth structure. For the evaluation of the results, three metrics were considered, i.e., Bit-Error-Rate (BER), Symbol-Error-Rate (SER) and Error-Vector-Magnitude (EVM). This work was compared with FPGA Codec-System (FCS) approach [28] which evaluated their work on Kodak dataset.

The results achieved by the FICH approach is presented in Figure 2, where the input image is presented in Figure 2 (a), compressed image is presented in Figure 2 (b), and decompressed image is presented in Figure 2 (c).


(a) Input (b) Compressed Image (c) Decompressed Image

(a) Input (b) Compressed Image (c) Decompressed Image

(a) Input (b) Compressed Image (c) Decompressed Image

Figure 2. Sample Output

Further, from the above results, the BER has been evaluated. The BER results achieved by FICH were compared with FCS approach and are presented in Figure 3. In Figure 3, the results are presented by considering small number of RRHs for transmission. From the results, it is seen that FICH achieved an average better BER improvement rate by 37.85% in comparison with FCS approach. Further, when considering more number of RRHs, the FICH approach achieved an average better BER improvement rate by 96.10%.

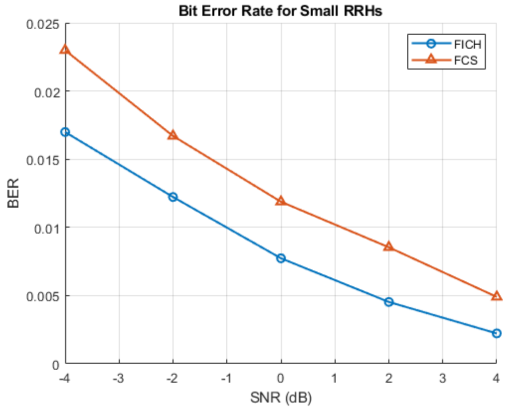


Figure 3. BER for less RRHs.

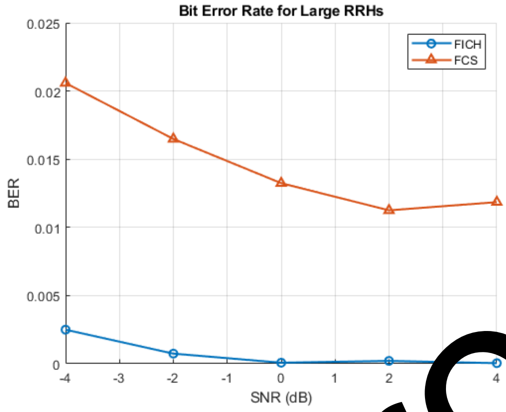


Figure 4. BER for more RRHs.

Further, the SER results achieved by the FICH approach are presented in Figure 5 and Figure 6 respectively. In Figure 5, the SER for considering small number of RRHs for transmission is evaluated where it was seen that the FICH approach achieved better SER improvement rate by 24.64%. In Figure 6, the SER was evaluated for large number of RRHs where it was seen that the FICH approach achieved better SER improvement rate by 91.13%. The FICH approach was also evaluated in terms of EVM. The results are presented in Figure 7 and Figure 8 for small and large number of RRHs respectivley. In Figure 7, the EVM for small number of RRH are evaluated where the FICH achieved an average better EVM improvement rate of 24.56%. In Figure 8, the EVM for large number of RRH are evaluated where the FICH achieved an average better EVM improvement rate of of 48.72%.

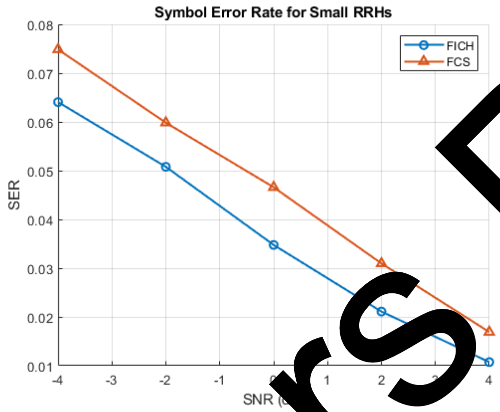


Figure 4. SER for less RRHs.

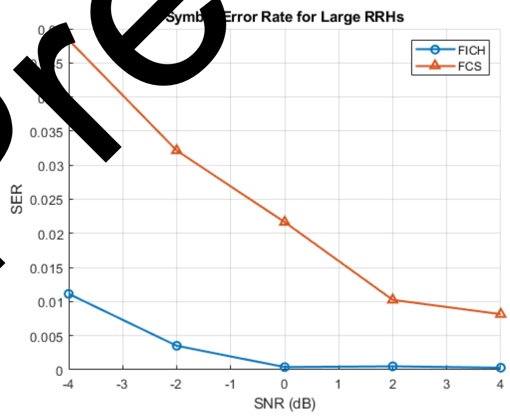


Figure 5. SER for more RRHs.

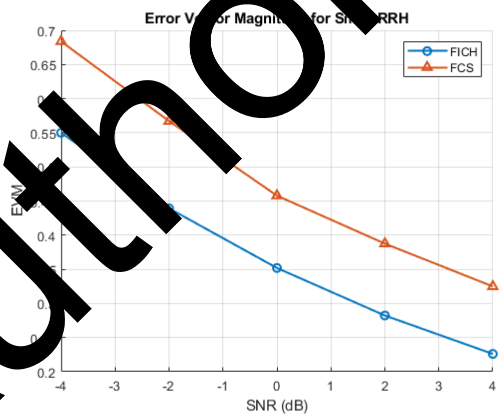


Figure 6. EVM for less RRHs.

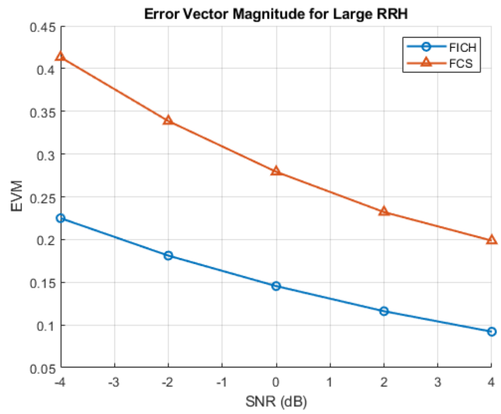


Figure 7. EVM for more RRHs.

The time taken for the execution of both the FICH and FCS approach is presented in Table 1 and presented graphically in Figure 8, Figure 9 and Figure 10. In this evaluation, we have evaluated 3 sequences of UVG dataset where encoding time and decoding time were evaluated. The findings from the table show that the FICH approach achieves faster image compression in comparison with existing FCS approach.

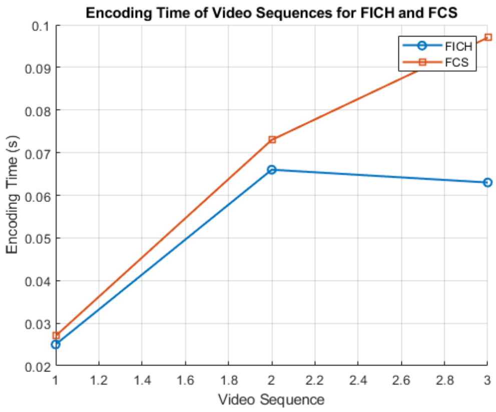


Figure 8. Encoding time for 3 sample video sequence.

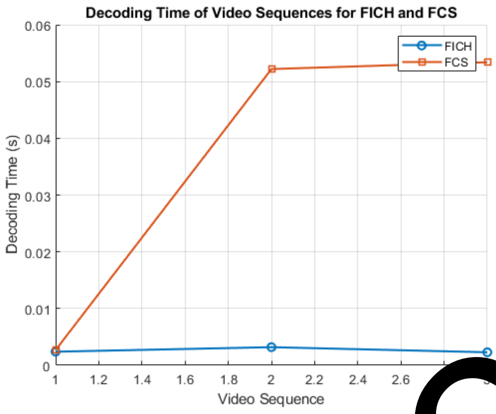


Figure 9. Decoding time for 3 sample video sequence.

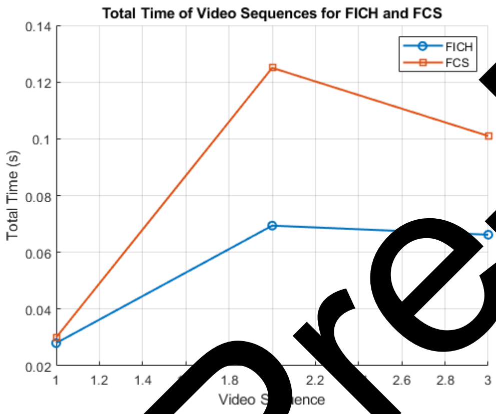


Figure 10. Total time evaluation for 3 sample video sequence.

Table 1. Encoding and Decoding time comparison with FCA.

Images	**Encoding Time**		**Decoding Time**		**Total Time**	
Video Sequence	**FICH**	**FCS**	**FICH**	**FCS**	**FICH**	**FCS**
1	0.025	0.027	0.0024	0.0027	0.028	0.03
2	0.066	[illegible]73	0.0032	0.0522	0.0694	0.125
3	0.[illegible]63	0.0[illegible]7	0.0023	0.0034	0.0662	0.101
AVG	0.0[illegible]333333	0.06[illegible]6667	0.002633333	0.0361	0.054533333	0.085333333

CONCLUSION

This research presented an FPGA-based image compression method utilizing the Huffman coding approach (FICH) aimed at enhancing the efficiency of wireless networks, specifically focusing on the Cloud-based Radio Access Network (C-RAN) architecture. The proposed FICH approach effectively utilized the low-rank estimation and Huffman coding techniques to compress image data efficiently, maintaining robustness against noise and errors. By leveraging FPGA-based hardware acceleration, the FICH approach optimized the use of computational resources, ensuring lower power consumption and higher throughput. This aligns with the growing demand for energy-efficient and high-performance solutions in next-generation wireless networks. The FICH method was developed to address the challenges of image compression within C-RAN, offering faster compression and decompression times compared to existing FPGA approaches. The FICH approach demonstrated significant improvements in Bit-Error-Rate (BER), Symbol-Error-Rate (SER), and Error-Vector Magnitude (EVM) when evaluated with both small and large numbers of Remote Radio Heads (RRHs). The method achieved an average BER improvement rate of 37.85% and 96.10% for fewer and more RRHs, respectively. Similarly, the SER and EVM improvements were substantial, with SER showing enhancements of 24.64% and 91.13%, and EVM improving by 24.56% and 48.72% for smaller and larger RRHs, respectively. The total execution time, including encoding and decoding processes, was notably reduced with the FICH approach. The average encoding and decoding times were significantly lower compared to the FCS approach, making the FICH method a faster alternative for real-time applications in wireless networks. Specifically, the average total time for image compression using FICH was 0.0545 seconds, compared to 0.0853 seconds for FCS, highlighting a clear improvement in efficiency. The findings of this study pave the way for more efficient and reliable wireless network infrastructures, supporting the increasing data demands of modern communication systems. In future work, the compressed image can be enhanced using DL approach.

REFERENCES

[1] A. Askri, C. Zhang and G. R. -B. Othman, "Distributed Learning Assisted Fronthaul Compression for Multi-Antenna C-RAN," in IEEE Access, vol. 9, pp. 113997-114007, 2021, doi: 10.1109/ACCESS.2021.3104660.

[2] J. P. Pavia, V. Velez, B. Brogueira, N. Souto, and A. Correia, “Precoded Generalized Spatial Modulation for Downlink MIMO Transmissions in Beyond 5G Networks,” Applied Sciences, vol. 10, no. 18, p. 6617, Sep. 2020, doi: 10.3390/app10186617.

[3] M. Rubina Aktar et al., "Energy-Efficient Hybrid Powered Cloud Radio Access Network (C-RAN) for 5G," in IEEE Access, vol. 11, pp. 3208-3220, 2023, doi: 10.1109/ACCESS.2023.3234190.

[4] R. T. Rodoshi, T. Kim, and W. Choi, “Resource Management in Cloud Radio Access Network: Conventional and New Approaches,” Sensors, vol. 20, no. 9, p. 2708, May 2020, doi: 10.3390/s20092708.

[5] D. Kumar, J. Kaleva and A. Tölli, "Blockage-Aware Reliable mmWave Access via Coordinated Multi-Point Connectivity," in IEEE Transactions on Wireless Communications, vol. 20, no. 7, pp. 4238-4253, July 2021, doi: 10.1109/TWC.2021.3057227.

[6] M. Polese, L. Bonati, S. D’Oro, S. Basagni and T. Melodia, "Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges," in IEEE Communications Surveys & Tutorials, vol. 25, no. 2, pp. 1376-1411, Secondquarter 2023, doi: 10.1109/COMST.2023.3239220.

[7] R. Dangi, P. Lalwani, G. Choudhary, I. You, and G. Pau, “Study and Investigation on 5G Technology: A Systematic Review,” Sensors (Basel, Switzerland), vol. 22, no. 1, p. 26, Dec. 2021, doi: 10.3390/s22010026.

[8] J. Kim and S.-H. Park, “Layered Inter-Cluster Cooperation Scheme for Backhaul-Constrained C-RAN Uplink Systems in the Presence of Inter-Cluster Interference,” Entropy, vol. 22, no. 5, p. 554, May 2020, doi: 10.3390/e22050554.

[9] O. Taghizadeh, T. Yang, H. Iimori, G. Abreu, A. C. Cirik and R. Mathar, "Quantization-Aided Secrecy: FD C-RAN Communications With Untrusted Radios," in IEEE Transactions on Wireless Communications, vol. 21, no. 10, pp. 8522-8535, Oct. 2022, doi: 10.1109/TWC.2022.3166865.

[10] Y. Zhou and W. Yu, "Fronthaul Compression and Transmit Beamforming Optimization for Multi-Antenna Uplink C-RAN," in IEEE Transactions on Signal Processing, vol. 64, no. 16, pp. 4138-4151, 15 Aug.15, 2016, doi: 10.1109/TSP.2016.2563388.

[11] N. M. Philip and N. M. Sivamangai, "Review of FPGA-Based Accelerators of Deep Convolutional Neural Networks," 2022 6th International Conference on Devices, Circuits and Systems (ICDCS), Coimbatore, India, 2022, pp. 183-189, doi: 10.1109/ICDCS54290.2022.9780683.

[12] D. Darwish, “Improving Techniques for Convolutional Neural Networks Performance,” European journal of electrical engineering and computer science, vol. 8, no. 1, pp. 1–16, Jan. 2024, doi: 10.24018/ejece.2024.8.1.596.

[13] Anouar Nechi, L. Groth, Saleh Mulhem, F. Merchant, Rainer Buchty, and Mladen Bereković, “FPGA-based Deep Learning Inference Accelerators: Where Are We Standing?,” ACM Transactions on Reconfigurable Technology and Systems, vol. 16, no. 4, pp. 1–32, Oct. 2023, doi: 10.1145/3613963.

[14] R. Schoonhoven, B. Veenboer, B. van Werkhoven and K. Batenburg, "Going green: optimizing GPUs for energy efficiency through model-steered auto-tuning," in 2022 IEEE/ACM International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS), Dallas, TX, USA, 2022 pp. 48-59, doi: 10.1109/PMBS56514.2022.00010

[15] S. Zhang, J. Cao, Q. Zhang, Q. Zhang, Y. Zhang and Y. Wang, "An FPGA-Based Reconfigurable CNN Accelerator for YOLO," 2020 IEEE 3rd International Conference on Electronics Technology (ICET), Chengdu, China, 2020, pp. 74-78, doi: 10.1109/ICET49382.2020.9119500.

[16] X. Yang, C. Zhuang, W. Feng, Z. Yang, and Q. Wang, “FPGA Implementation of a Deep Learning Acceleration Core Architecture for Image Target Detection,” Applied sciences, vol. 13, no. 7, pp. 4144–4144, Mar. 2023, doi: 10.3390/app13074144.

[17] X. Wang et al., “An FPGA-based online reconfigurable CNN edge computing device for object detection,” Microelectronics Journal, vol. 137, pp. 105805–105805, Jul. 2023, doi: 10.1016/j.mejo.2023.105805.

[18] R. Kumar, S. K. Gupta, H.-C. Wang, C. S. Kumari, and S. S. V. P. Korlam, “From Efficiency to Sustainability: Exploring the Potential of 6G for a Greener Future,” Sustainability, vol. 15, no. 23, p. 16387, Jan. 2023, doi: 10.3390/su152316387.

[19] M. Zhang, L. Li, H. Wang, Y. Liu, H. Qin, and W. Zhao, “Optimized Compression for Implementing Convolutional Neural Networks on FPGA,” Electronics, vol. 8, no. 3, p. 295, Mar. 2019, doi: 10.3390/electronics8030295.

[20] Y. Barrios et al., “Lossy Hyperspectral Image Compression on a Reconfigurable and Fault-Tolerant FPGA-Based Adaptive Computing Platform,” Electronics, vol. 9, no. 10, p. 1576, Oct. 2020, doi: 10.3390/electronics9101576.

[21] M. Ledwon, B. F. Cockburn and J. Han, "High-Throughput FPGA-Based Hardware Accelerators for Deflate Compression and Decompression Using High-Level Synthesis," in IEEE Access, vol. 8, pp. 62207-62217, 2020, doi: 10.1109/ACCESS.2020.2984191.

[22] S. Jang, W. Liu, and Y. Cho, “Convolutional Neural Network Model Compression Method for Software—Hardware Co-Design,” Information, vol. 13, no. 10, p. 451, Oct. 2022, doi: 10.3390/info13100451.

[23] K Pranitha and G Kavya, “An efficient image compression architecture based on optimized 9/7 wavelet transform with hybrid post processing and entropy encoder module,” Microprocessors and Microsystems, vol. 98, pp. 104821–104821, Apr. 2023, doi: 10.1016/j.micpro.2023.104821.

[24] T. Ali, D. Bhowmik, and R. Nicol, "Domain-Specific Optimisations for Image Processing on FPGAs," Journal of signal processing systems for signal, image, and video technology, vol. 95, no. 10, pp. 1167–1179, Sep. 2023, doi: 10.1007/s11265-023-01888-2.

[25] Y. Su, Kah Phooi Seng, J. Smith, and Li Minn Ang, "Efficient FPGA Binary Neural Network Architecture for Image Super-Resolution," Electronics, vol. 13, no. 2, pp. 266–266, Jan. 2024, doi: 10.3390/electronics13020266.