Available online at website: https://jurnal.iaii.or.id/index.php/RESTI

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 10 No. 5 (2026) 1393 - 1403 e-ISSN: 2580-0760

Modified Quantization Strategy for Enhanced Lossless RAW Audio Compression

Tonny Hidayat[1], Hendra Kurniawan[2], Hastari Utama[3], Dewi Anisa Istiqomah[4], Joko Dwi Santoso[5]
[1,2,3,4,5]Faculty of Computer Science, Universitas Amikom Yogyakarta, Yogyakarta

[*]*tonny@amikom.ac.id*

Abstract

The amount of audio data required in today's world is rapidly expanding, posing a significant problem and challenge in terms of media storage and transfer. RAW audio data compression has grown in popularity due to its ability to lower file size without sacrificing quality. To anticipate the complexity of increasing the lossless audio compression ratio, this study proposes a lossless audio data compression algorithm that theoretically increases its efficiency ratio, thus providing multiple compression that combines linear prediction methods, integer wavelet transforms, and as a back-end process using Huffman code for binary output coefficients. The combined transformation produced employs a new prediction form and coefficient process, which is a novelty and the primary contribution of the coding of this algorithm structure. Based on the test results in many audio categories, the obtained results demonstrate that the proposed algorithm surpasses the performance of previous studies. The compression ratio increased from 2% to 5%. The algorithm discovered in this work is ideal for compressing audio with complex patterns and high irregularity using a quick compression procedure.

Keywords: binary; compression; huffman; lossless

How to Cite: T. Hidayat, H. Kurniawan, H. Utama, D. Anisa Istiqomah, and J. Dwi Santoso, "Modified Quantization Strategy for Enhanced Lossless RAW Audio Compression", *J. RESTI (Rekayasa Sist. Teknol. Inf.)* , vol. 10, no. 5, pp. 1393–1403, Oct. 2026, doi: 10.29207/resti.v10i5.7512.
Permalink/DOI: https://doi.org/10.29207/resti.v10i5.7512

Received: January 22, 2026
Accepted: September 5, 2026
Available Online: October 9, 2026

This is an open-access article under the CC BY 4.0 License
Published by Ikatan Ahli Informatika Indonesia

1. Introduction

The current demand for bandwidth and storage media is extremely high, particularly due to the massive size of Hi-Res RAW audio files. Audio compression is widely utilized in a variety of applications, including streaming, audio on demand, television broadcasting, teleconferencing, and telecommunications. The development of compression techniques presents a significant production issue. There are two forms of compression: lossless and lossy [1]. Lossless compression generates files that can be decompressed back to their original format.

In contrast, lossy compression causes audio to lose some of its original nuances after compression. Lossless audio compression is commonly employed for high-fidelity applications, such as audio masters in music and film studios [2], satellite imagery communications [3], remote sensing, and digital forensic data [4]. As an example, in the case of film, audio quality has a significant impact on the mood while viewing a film; the details of the sound created will provide satisfaction to each scene that occurs. To get that sensation, audio must be mastered in a lossless format that allows every feature of the audio instrument to be heard properly. A proficient lossless compression approach can increase the delight of audiovisual enthusiasts and manufacturers.

A law was developed based on Claude E. Shannon's probability theory, which states that data can be compressed based on its probability [5]. The redundancy value dictates how efficiently the compression ratio may be created. This is also consistent with David A. Huffman's hypothesis, which states that the lower the redundancy value, the higher the ratio produced [6]. There has been some relevant study into lossless audio compression techniques [7]. The most recent audio compression technologies include entropy coding [8], run-length encoding (RLE) [9], and dictionary-based coding [10].

Entropy coding, such as Shannon-Fano, Huffman, and arithmetic coding, produces better audio compression than RLE because the RLE system stores data as a

single value and sorts it by the number of identical values, resulting in a substantially shorter code length. As a result, RLE coding is effective for data containing a large number of comparable data symbols. Unlike the entropy method, the probability of audio data is ordered from largest to smallest and then divided into two groups with nearly equal numbers in each. The preceding steps will continue until each element is a node on each branch of the binary tree. Shannon-Fano coding cannot always create correct data and has a low compression ratio since it is unable to form an optimized binary tree. Huffman coding improves on this approach by sorting the binary tree form of Huffman code from branch to root, which is the reverse of Shannon-Fano sorting [11]. Huffman coding yields a more optimum binary structure [12], although it is highly sensitive to noise. Arithmetic coding converts input into a static decimal range based on a predetermined probability set and then keeps just the top and lowest limits [13].

Arithmetic compression produces better results, but it is quite sluggish in terms of time compression, and a bit error in arithmetic coding might destroy the audio stream. Dictionary-based coding, which uses a lookup table to map a code to a sequence of characters, is another way of lossless compression. Lempel-Ziv-Welch (LZW) is a universal lossless compression technique, with LZ77 and LZ78 being the most recent versions as of the writing of this paper. LZW creates a list of entries that represent various probabilities of recurrence for strings that are constantly repeated [14] . This scheme is suitable for estimating the size of files with similar repeating probability, but it is more complicated and requires a significant amount of processing time if the file is huge and continuous [15].

The current tendency is to combine numerous coding systems, such as Canonic, Deflate, and CALIC, which are frequently used and suggested. Canonic is an inverse approach that is frequently employed in the standard algorithm for continuous-stream audio compression without loss and with minimal data loss and corruption [16]. This standard is based on a variant of the low complexity lossless compression method (LOCO-I) [17], and by combining the context model with Golomb coding, the resulting compression ratio outperforms other related schemes in the report [18]. EBCOT (Embedded Block Coding with Optimized Truncation of the embedded bitstreams) is the next step in this coding, and it currently outperforms others in compression performance [19]. Deflate is a combination of the LZ77 algorithm and the Huffman code, and the result is claimed to be more efficient than LZW in terms of compression ratio and time compression [20]. The context-based, adaptive, lossless image codec (CALIC) is a sequential coding technique that combines context modelling and arithmetic coding. CALIC is one of the most efficient lossless compression algorithms, having a superior compression ratio [21].

One of the most common methods for improving compression without losing data variational information is to reconstruct the binary structure. Metric indexes and spectral maps define the indexed audio waveform. Each index corresponds to a place in the waveform data. This restructuring tries to increase the lossless compression rate by determining the best permutation of the waveform signal data map [22]. Based on Oliver Giudice et al.'s proposal of a new re-indexing methodology in which the traveling salesman problem is handled using ant colony optimization (TSP-ACO) [23], this method outperforms the current method in terms of compression ratio. Singhai and Kumar improved TSP-ACO by applying a rotational shift to the best signal waveform map for a given spectrum following ant colony optimization [24], which improved the compression outcomes of the previous innovative approach.

Recent scientific research on deep learning has become more deeply embedded in lossless audio compression. A new CNN-based prediction paradigm in [25] is specifically developed to anticipate high-texture regions in RAW/Hi-Res audio. Wu et al. describe a CNN-based predictive lossless compression strategy for multi-instrumented waveform audio [26], achieving exceptional lossless compression performance on RAW audio from a microphone. However, these learning-based approaches employ deep neural networks - PixelCNN++ [27], and Multiscale-Pixel CNN [28], which always demand more processing time on GPU with high computational complexity and still necessitate an entropy coding process in the final step [29].

The most recent method approach for lossless audio compression combines the two-dimensional discrete wavelet transform (DWT) and the one-dimensional discrete fractional Fourier transform (1D-DFrFT) [30]. They divide the likelihood of data into low- and high-frequency sub-bands. These two types of sub-bands are compressed using distinct approaches. The ensuing compression process demonstrates that the approach achieves a greater compression ratio while reconstructing audio data bits more accurately.

Azman et al introduced hybrid predictive strategies in the DPCM-IWT-Huffman structure to improve the performance of lossless compression algorithms [31]. Yet since the prediction formulation and Haar wavelet are constrained and tend to be simple, some spatial and spectral redundancy may be reduced further. As a result of these constraints, obtaining the best entropy and compression ratio outcomes becomes difficult.

The lossless compression method's primary purpose is to optimize and improve the compression ratio; however, time compression must also be addressed. Many existing compression algorithm techniques have complex architecture, resulting in limited ratio improvement. This work proposes a lossless algorithm coding technique with compression performance capable of boosting the compression ratio while

maintaining a basic structure. The key addition of this algorithm scheme is the novel hybrid transformation, which makes use of the new prediction structure and IWT coefficient processing. The results of the study will be given, demonstrating that the algorithm that will be constructed will improve the coding efficiency of lossless audio compression while also reducing the technique's complexity. The wavelet transform will be used to predict errors, reducing data redundancy. Added a process to remove zero and negative values from wavelet coefficients. Finally, we will provide a comparison with different lossless compression algorithms.

Unlike previous hybrid approaches such as Azman et al. [31] who combined DPCM, IWT, and Huffman with limited prediction formulation and simple Haar wavelet, the proposed Modified Quantization (MQ) method introduces three key novelties. First, it employs a 2-channel linear predictor specifically optimized for audio waveform redundancy, which produces smaller and more centrally distributed error values compared to conventional DPCM. Second, it applies a coefficient processing stage after integer wavelet transformation that explicitly removes zero and negative values, thereby increasing symbol probability for the entropy encoder. Third, the integration of these stages is designed to maintain low computational complexity while achieving a superior compression ratio, as demonstrated in Section 3. These novelties collectively enable MQ to outperform existing lossless audio compression methods on high-redundancy audio categories while maintaining competitive performance on low-redundancy signals.

2. Methods

This section will discuss the complexity of the developed algorithm scheme, which combines linear predictive coding, integer wavelet transformation as coefficient processing, and the Huffman code method for back-end processing. The flow diagram of the encoder in the designed technique is shown in Figure 1. The audio channel is entered first, followed by a linear prediction to acquire the integer error string symbol due to the large and close correlation. The predictor's error value is significantly lowered, approaching zero. This significantly assists the wavelet transform in the subsequent stage, resulting in a more concentrated coefficient distribution. Figure 1 illustrates the overall encoding flow of the proposed MQ method, starting from raw audio input to compressed binary output.

In the following phase, IWT is applied to the error prediction. The observations show that the IWT output coefficients are relatively small, with values concentrated at zero or close to zero. Under this condition, the IWT output coefficients are processed using a technique that removes zero and negative values and records their positions in the data array. Mapping and recording are critical because accurately recreating data requires exact information, particularly for lossless compression. The audio data is then correctly processed to reduce its entropy, increase the efficiency of entropy coding, and finally encoded entropically using the Huffman code algorithm scheme to produce a stream of binary digits that are arranged and compressed.

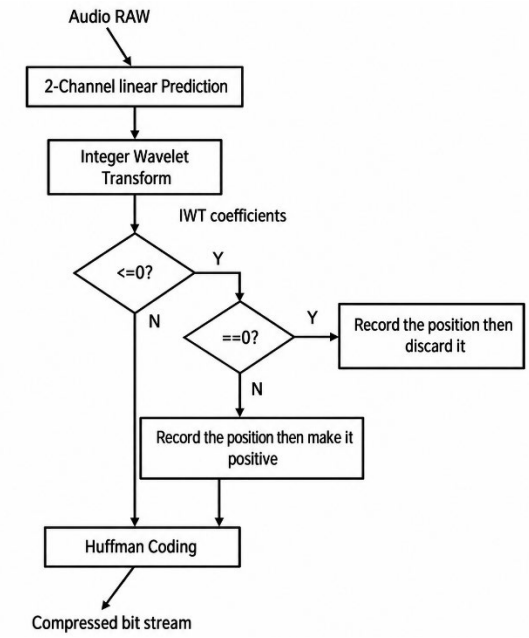


Figure 1. Flowchart of Method Design Process

2.1 Lossless Linear Channel Prediction

Predictor optimization and the ability to remove most spatial redundancy make this stage crucial for the efficiency of the compression process. The potential to realize integer systems is greatly enhanced by linear predictors [32]. By applying the benefits of the 2-channel linear predictor, audio symbol redundancy was eliminated. The predicted value will be added to an integer and then added once more to the closest integer in order to perform transformation without causing harm or loss. This will result in an integer representing the difference between the original value and the predicted value. This structure ensures that the reconstructed audio is identical to the original. Figure 2 depicts a block schematic of the lossless linear predictive encoding and decoding structure.

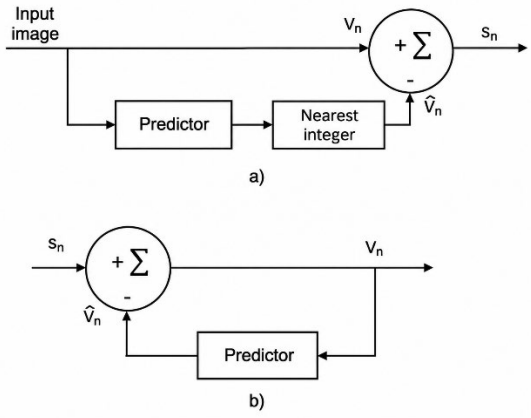


Figure 2. Prediction Procedure: Encoder (a) and Decoder (b).

The encoder subtracts the rounded predicted value from the original sample to obtain the error value, while the decoder reconstructs the original sample by adding the error back to the predicted value. This procedure is illustrated in Figure 2. The upper component is transmitted through the predictor, while the component at the symbol frequency of the next channel is calculated using the defined prediction template. The acquired data is then rounded to the nearest expected value V¯n. In the end, the integer predicted value fn is subtracted from the original audio channel value V¯n to obtain the error value Sn. The reversing process of our encoding is called decoding, and the prediction framework is shown in Figure 3.

$x_n * h_n$	$x_n * w_n$	$x_n * y_n$
$\hat{X}_n * H_n$	$\hat{X}_n * W_n$	$\hat{X}_n * Y_n$

Figure 3. Mapping Value Bitstream

Figure 3 shows the 5-point prediction template used in Equation (1), where the predicted value $F_{(x,y)}$ is calculated from five neighboring samples. $K_{(x,y)}$ in Figure 3 shows a representation of the audio bit stream value at position *(x,y)*. At this point, the audio input's middle waveform value is retained, while the middle waveform values at other positions are added to yield the anticipated value $F_{(x,y)}$, as shown in Equation 1. Equation 2 continues by rounding the projected value. The following phase will create an error value $S_{(x,\ y)}$ obtained from Equation 3, by decreasing the original waveform $K_{(x,y)}$ with the stipulation that the predicted value is rounded $\bar{F}_{(x,y)}$, this computation ensures that the data is not lost and remains accurate.

$$F_{(x,y)} = \frac{1}{5}\left(K_{(x-1,y-1)} + K_{(x-1,y)} + K_{(x-1,y+1)} + K_{(x,y-1)} + K_{(x,y+1)}\right) \quad (1)$$

$$\overline{F_{(x,y)}} = round(F_{(x,y)}) \quad (2)$$

$$S_{(x,y)} = K_{(x,y)} - F_{(x,y)} \quad (3)$$

2.2 Integer Wavelet Transform

Lifting and level modifications to the upper position by Pramanik Sabyasachi [33] result in a wavelet transform from integer to integer with a reversible character. The basic technique for this transformation consists of three stages, as shown in Figure 4. The first step is to divide the input signal into two channels with odd and even sample values [34], [35]. In the second step, prediction, the predictor will use even-valued samples $c_{r-1,g}=d_{r,2g}$ to forecast odd-valued samples $P_{r-1,g}=d_{r,2g+1}$, and the anticipated outcomes mr-1 will be expressed as high-frequency coefficients. The third procedure, known as the sample merging update phase for new values, the even sample value cr-1 is updated with mr-1 to generate the low frequency coefficient value dr-1 in the prediction process, and the update is performed in reverse order during the transformation process. As a result, two distinct sets of frequencies will be blended into a single channel path to produce the optimal data bit reconstruction data [36].

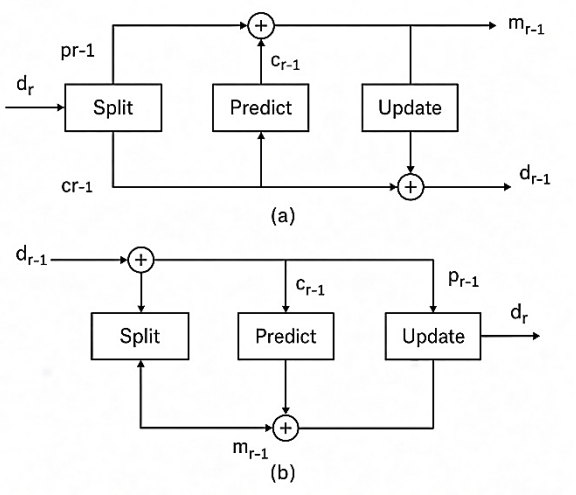


Figure 4. Lifting scheme for Integer Wavelet Transform (IWT)

Selecting the right wavelet is required in order to adhere to the lossless compression rule, which states that nothing should be destroyed or lost. The decision is based on LeGall 5/3 technique, which enables transformation from integer to integer. LeGall 5/3 Wavelet can be used to support the biorthogonal wavelet with two vanishing points [37]. This step effectively removes the correlation between audio waveforms in a brief amount of time. In order to prevent phase distortion during audio processing, the filter is symmetrical. This approach serves as evidence that the audio data source reconstruction method is accurate.

The forward transform version of the 5/3 filter can be found in Equations 4 and 5. Where dr is the error value *F* acquired from the predictive coding step, m_{r-1} and d_{r-1} are the high and low frequency sub-bands after the wavelet transform, respectively, at each frequency. To optimize compression performance, six levels of wavelet decomposition will be performed as {dr-6, mr-6, mr-5, …, mr-1}.

$$m_{r-1,g} = d_{r,2g+1} - \left\lfloor \frac{d_{r,2} + d_{r,2g+2}}{2} \right\rfloor \quad (4)$$

$$d_{r-1,g} = d_{r,2g} + \left\lfloor \frac{m_{r,2} + m_{r,2g+2}}{4} \right\rfloor \quad (5)$$

The inverse transform of the integer wavelet LeGall is defined in Equations 6 and 7.

$$d_{r,2\,g} = d_{r-1,g} - \left\lfloor \frac{m_{r-1,g-1} + m_{r-1,g+1} + 2}{4} \right\rfloor \quad (6)$$

$$d_{r,2g+1} = m_{r-1,g} + \left\lfloor \frac{d_{r,2} + d_{r,2g+2}}{2} \right\rfloor \quad (7)$$

The inverse transformation is defined during the decompression process, where dr,2g and dr,2g+1 representing the original data's even and odd sub sequences dr. Le Gall's limitations throughout the integer wavelet translation process include the ability to conduct only subtraction and addition [38]. It will lose precision in the rounding procedure during both the

forward and reverse transformations. In other cases, the same quantity will be lowered and increased in the reverse transformation if there is more or less in the forward transformation. The lossless property is used in this section, which means that nothing is harmed or lost. Errors occurring before or after the transformation must equal zero in order for lossless compression to be achieved; this is done after the rounding process

2.3 Coefficient Process

This is a simulation of the wavelet coefficient. This process is carried out following the predictor and IWT on the audio data, as an audio sample can be seen in Figure 5 in orange, the signal waveform is seen in the bitstream histogram as in Figure 5 in blue, and the predictor's output shape is similar to Figure 6. The output process of the IWT predictor is depicted in Figure 7, respectively. When comparing Figure 5 to the others, it is clear that the value of 0 is quite dominant and increases symmetrically. Therefore, this method serves as the foundation for presenting a procedure for processing the wavelet coefficient; below is the resulting rule.

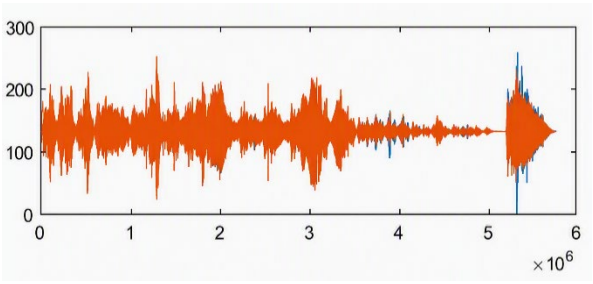


Figure 5. Original Waveform (Orange) with Histogram (blue).

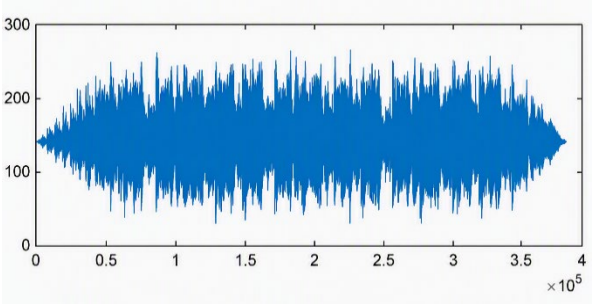


Figure 6. Linear Prediction

The coefficient processing stage applies three sequential rules to each wavelet coefficient generated by the Integer Wavelet Transform. First, if the value of a wavelet coefficient is zero, its position in the array *x(i)* is recorded and set to zero, and the coefficient is subsequently omitted from further processing. Second, if the coefficient is negative, its position in the array *y(i)* is recorded, and its absolute value is retained for addition or subtraction in the subsequent stage. Third, if the resulting coefficient value is positive, it is kept unchanged and held for the following encoding process. These three rules collectively establish a procedure that enhances the probability distribution of the data, thereby improving the compression ratio and reducing data redundancy in the entropy-based coding stage that follows.

This rule establishes a procedure for enhancing the probability of data. The main consequence is visible in the indicator of appropriate compression ratio and reduced data redundancy, which only applies to entropy-based coding. Figures 5, 6, and 7 show the original waveform histogram, linear prediction output, and IWT coefficient distribution respectively, demonstrating how the data becomes increasingly concentrated around zero.

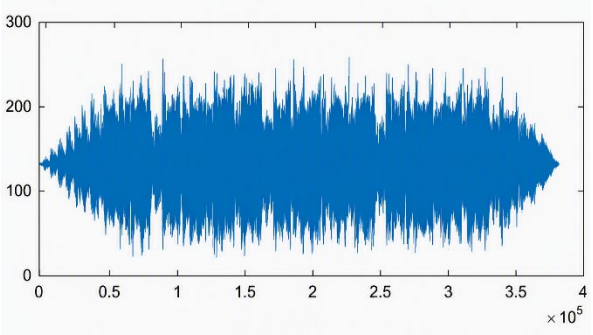


Figure 7. Waveform Wavelet Transformation

2.4 Huffman Coding

With the complexity of the algorithm and the effect of optimal compression than other algorithms. Huffman code can offer a better and more intuitive substitute due to its complexity and excellent compression impact compared to other algorithms [39]. Therefore, the altered data is encoded using the Huffman code's entropy coding technique in order to get the best compression ratio. These are the provisions of the back-end process that this study uses to apply the Huffman coding procedure to the lossless compression strategy.

The Huffman coding procedure applied in this study consists of four sequential steps. First, the symbol variants and the number of data probabilities are calculated and sorted in descending order, from the largest to the smallest probability. Second, the construction of the binary tree begins at the root with two branches that start from the symbol variation with the smallest number of probabilities. Third, new branches or roots are formed by combining the values of the two smallest probabilities, and this merging process is repeated continuously until all branches converge into a single root node. Fourth, binary digits 0 and 1 are assigned to each left and right branch of the resulting binary tree generated by the Huffman code, thereby producing the final prefix-free codewords used to encode the processed coefficient stream.

2.5 Performance Metrics

Two metrics are used to evaluate the proposed lossless compression method. The first is the compression ratio (CR), defined as the ratio of the compressed file size (TC) to the total length of the original code (TL), expressed as a percentage shown in Equation 8.

$$CR = \frac{TC}{TL} \times 100\% \quad (8)$$

A lower CR indicates better compression performance. The second metric is compression time (CT), measured in megabytes per second (MBps), where a higher value indicates faster compression.

2.6 Proposed Algorithm

The complete encoding procedure of the proposed Modified Quantization (MQ) method is described in the following three phases. The first phase implements the linear channel prediction described in Section 2.1. For each audio sample $K(x, y)$in the input channel, the algorithm computes a predicted value $F(x, y)$based on five neighboring samples using Equation 1. This predicted value is then rounded to the nearest integer using Equation 2 to obtain $\bar{F}_{(x,y)}$. The prediction error $S(x, y)$is subsequently calculated by subtracting the rounded predicted value from the original sample using Equation 3, producing a small-magnitude error value that has lower entropy than the original signal.

The second phase performs the Integer Wavelet Transform and coefficient processing as described in Sections 2.2 and 2.3. The prediction errors are first transformed using the LeGall 5/3 integer wavelet filter, which decomposes the signal into low-frequency and high-frequency sub-bands according to Equations 4 and 5. Following the transformation, the coefficient processing stage explicitly handles zero and negative values. When a wavelet coefficient equals zero, its position is marked and the coefficient is discarded from further processing. When the coefficient is negative, its absolute value is stored and its position is marked as negative. When the coefficient is positive, it is retained without modification. This three-rule processing strategy significantly concentrates the data distribution around dominant symbols, thereby improving the efficiency of the entropy coding stage.

The third phase applies Huffman coding as described in Section 2.4. The processed coefficient stream, which now contains fewer symbols and a more concentrated probability distribution, is encoded using a Huffman tree. The tree is constructed by sorting symbols according to their probabilities, repeatedly merging the two smallest probabilities until a single root node is formed, and assigning binary digits 0 and 1 to each branch. The resulting binary stream is returned as the final compressed output. The complete decoding process follows the reverse order: Huffman decoding, coefficient reconstruction to restore zero and negative values using the recorded position information, inverse IWT using Equations 6 and 7, and finally addition of the predicted values back to the error values to reconstruct the original audio signal losslessly.

3. Results and Discussions

This section presents the experimental evaluation of the proposed Modified Quantization (MQ) lossless audio compression method. The evaluation is conducted on a large and diverse dataset comprising 9,000 audio files from five public corpora, with technical specifications including sampling rates ranging from 8 kHz to 192 kHz, bit depths of 16 and 24 bits, and both mono and stereo channel configurations. Two primary performance metrics are used to assess the proposed method: compression ratio (CR), measured as the percentage of the original file size after compression, where a lower value indicates better performance; and compression time (CT), measured in megabytes per second (MBps), where a higher value indicates faster compression.

The proposed MQ method is compared against six established lossless compression schemes, namely the standard Huffman algorithm, Dynamic Huffman algorithm, WinZip, 7-Zip, PkZip, and Monkey's Audio. The following subsections describe the dataset in detail, present the comparative results for compression ratio and compression time, discuss the findings in relation to the algorithmic innovations introduced in Section 2, and finally address the limitations of the current study along with directions for future research.

3.1 Dataset Description

To ensure the robustness and generalizability of the proposed Modified Quantization (MQ) lossless compression method, a large and diverse dataset comprising 9,000 audio files was compiled from five publicly available corpora [40-44]. The dataset was originally categorized into eight perceptual types: music, music mono, music stereo, ripping CD, speech, noise, sound effects, and instruments. While this taxonomic classification ensures diversity in audio content, a technical characterization based on digital parameters is essential for evaluating compression performance across different temporal resolutions.

Therefore, Table 1 presents a refined categorization of the dataset based on sampling rate (fs), bit depth, and channel configuration. This addresses the critical need to assess how the proposed method handles both sparse data (low fs) and dense data (high fs).

Defined qualitatively based on the correlation between adjacent samples. Higher sampling rates (e.g., 96 kHz, 192 kHz) produce higher temporal redundancy, which is theoretically more favorable for lossless compression.

To further substantiate the claim of enhanced lossless compression, three design choices were implemented in the dataset construction. The first was the inclusion of high-resolution audio, in which subsets with 96 kHz and 192 kHz sampling rates were added, totaling 1,200 files. This is important because lossless compression typically achieves its best compression ratio at high sampling rates due to the strong correlation between consecutive samples. The second design choice was the standardization of speech data, in which the speech category was split into 8 kHz, representing VoIP or telephony quality, and 16 kHz, representing wideband quality. This split allows the robustness of the proposed

method to be tested on low-redundancy data, where prediction errors are naturally larger. The third design choice was the resolution of conceptual overlap, in which categories such as "Mono" and "Stereo" are treated as channel attributes within the technical table rather than as separate content categories. This avoids double-counting and ensures a fair comparison across algorithms.

Table 2 summarizes the five public corpora used in this study, including their abbreviations, main content types, total file contributions, and access links.

Table 1. Technical Specification of the Dataset by Sampling Rate and Content Type

Content Category	Number of Files	Sampling Rate (kHz)	Bit Depth	Channel	Redundancy Level	Primary Source
Speech (VoIP)	600	8	16	Mono	Low	MUSAN [40]
Speech (Wideband)	600	16	16	Mono	Medium-Low	MUSAN [40]
Music (CD)	1,800	44.1	16	Stereo	Medium	Whittle [44]
Music (Hi-Res)	900	96	24	Stereo	High	IRMAS [42]
Music (Hi-Res)	300	192	24	Stereo	Very High	IRMAS [42]
Noise	1,200	44.1	16	Mono	Low	MUSAN, FSD [43]
Sound Effects	1,500	44.1 / 48	16	Stereo	Medium	FSDnoisy18k [43]
Instruments	1,800	44.1	16	Stereo	High	IRMAS [42]
Ripping CD	900	44.1	16	Stereo	Medium	Whittle [44]
Total	9,000	-	-	-	-	-

Table 2. Summary of Public Audio Corpora Used in the Dataset

No	Corpus Name	Abbreviation	Main Content Types	File Contribution	Access Link
1	MUSAN: A Music, Speech, and Noise Corpus	MUSAN [40]	Music, speech, noise	~2,500	http://www.openslr.org/17/
2	Pitch estimation for melody extraction (Bosch et al.)	Bosch2016 [41]	Symphonic classical music	~1,500	https://zenodo.org/record/1289786
3	IRMAS Instrument Recognition Dataset	IRMAS [42]	Musical instruments, sound segregation	~2,000	https://zenodo.org/record/1290750
4	FSDnoisy18k	FSDnoisy18k [43]	Sound events with noisy labels	~1,800	https://zenodo.org/record/2529934
5	Lossless Audio Compression CD Quality (Whittle)	Whittle [44]	CD-quality music tracks	~1,200	http://www.firstpr.com.au/audiocomp/lossless/
Total		-	-	9,000	-

The first corpus, MUSAN [40], provides a balanced collection of music, speech, and noise samples, making it suitable for evaluating compression performance across fundamentally different signal characteristics. The second and third corpora, Bosch2016 [41] and IRMAS [42], focus on melodic and instrumental audio, respectively. These are particularly useful for testing the proposed MQ method on complex, polyphonic signals with high spectral variability. The fourth corpus, FSDnoisy18k [43], introduces real-world noisy conditions and sound events, which challenge the robustness of the prediction and wavelet transform stages. Finally, the Whittle corpus [44] provides standard CD-quality audio (44.1 kHz, 16-bit stereo), serving as a benchmark for comparing MQ against legacy lossless compression schemes. By aggregating these five distinct sources, the final dataset achieves a total of 9,000 files, ensuring statistical credibility and diversity across both content types and acoustic complexities.

3.2 Compression and Ratio Results

Based on the techniques and criteria outlined in Section 2, we evaluated the lossless compression performance of the proposed MQ scheme against six state-of-the-art lossless compression methods: Huffman algorithm, Dynamic Huffman algorithm, WinZip, 7-Zip, PkZip, and Monkey's Audio [46]. The evaluation was conducted across eight audio categories.

Table 3 presents the compression ratio (CR) results, where a lower percentage indicates better compression performance. All values are rounded to three decimal places for consistent comparison. To clearly demonstrate the superiority of the proposed MQ method, Δ (difference) columns are added next to each comparison method, showing the difference between that method's CR and MQ's CR. Positive Δ values indicate MQ achieves a lower (better) CR than the comparison method; negative Δ values indicate the comparison method outperforms MQ for that specific audio type. As shown in Table 3, the proposed MQ method achieves the best compression ratio in 5 out of 8 audio categories (music, music mono, music stereo, sound effect, and instruments). For *music stereo*, MQ achieves 37.653%, outperforming WinZip (38.094%) by 0.441 percentage points. For *sound effect*, MQ achieves 26.939%, which is 1.341 percentage points better than WinZip (28.280%). The average CR across all categories for MQ is 38.509%, which is superior to Huffman (40.393%), Dynamic (38.922%), WinZip (39.050%), 7-Zip (42.163%), and Monkey's Audio (45.114%). Notably, MQ performs exceptionally well on complex, high-redundancy audio such as *instruments* (40.565%) and *music stereo* (37.653%). However, for *ripping CD* and *speech*, MQ is slightly outperformed by WinZip (Δ = -0.321%) and Huffman (Δ = -0.427%), respectively, indicating opportunities for further improvement.

Table 3. Compression Ratio Results of the Dataset by Sampling Rate and Content Type

Audio Type	Huffman	Δ vs MQ	Dynamic	Δ vs MQ	WinZip	Δ vs MQ	7-Zip	Δ vs MQ	Monkeys	Δ vs MQ	MQ
Music	47.187	+2.937	46.884	+2.634	44.666	+0.416	47.263	+3.013	50.658	+6.408	44.250
Music mono	39.515	+3.805	35.360	-0.350	36.230	+0.520	39.240	+3.530	42.230	+6.520	35.710
Music stereo	41.334	+3.681	40.435	+2.782	38.094	+0.441	41.311	+3.658	44.206	+6.553	37.653
Ripping CD	42.858	-5.225	41.769	-6.314	47.762	-0.321	50.834	+2.751	54.351	+6.268	48.083
Speech	38.640	-0.427	37.908	-1.159	39.666	+0.599	42.735	+3.668	45.678	+6.611	39.067
Noise	39.002	+3.195	35.472	-0.335	36.714	+0.907	39.891	+4.085	42.451	+6.644	35.807
Sound effect	30.106	+3.167	29.628	+2.689	28.280	+1.341	31.863	+4.924	34.519	+7.580	26.939
Instruments	44.502	+3.937	43.922	+3.357	40.990	+0.425	44.170	+3.605	46.818	+6.253	40.565
AVERAGE	40.393	+1.884	38.922	+0.413	39.050	+0.541	42.163	+3.654	45.114	+6.605	38.509

3.2 Compression Time Results

Aside from the ratio, the length of the compression process is the most important metric in lossless compression. The Compression Time (CT) is the time it takes to compress file i (in megabytes per second - MBps), as shown in (9). i represents each sort of audio (Music, Music Mono, Music Stereo, Ripping CD, Speech, Noise, Sound Effects, Instruments). Because each file is large, this calculation employs MB (mega bytes). So, the computation is File size (UF) divided by compression time (seconds). So, the largest number is the fastest time obtained by the compression procedure as shown in Equation 9.

$$CT_i = \frac{UF}{t} \quad (9)$$

Table 4 shows that some component versions have CT advantages depending on the audio format. The MQ approach provides CT benefits in four categories of audio: music mono (9.903 MBps), music stereo (5.788177 MBps), sound effects (4.977084 MBps), and instruments (4.709709 MBps). 7-Zip (4.72169 MBps) is the best CT for music audio, and Huffman (3.743784 MBps) is the greatest for CD ripping. WinZip has the best Ct for speech and noise audio types, with 4.6162 MBps and 3.029888 MBps, respectively. Looking at the total average of this compression time procedure, it can be observed that MQ has the highest average (5.066985 MBps).

Table 4. Compression Time Results (MBps)

Audio Type	Huffman	Dynamic	WinZip	7-Zip	Monkeys	MQ
Music	4.464	0.075	4.574	4.722	4.287	4.577
Music mono	5.661	0.144	5.905	5.728	6.717	9.903
Music stereo	5.752	0.115	5.747	5.290	5.450	5.788
Ripping CD	3.744	0.105	3.343	3.620	3.631	3.292
Speech	4.306	0.138	4.616	4.095	4.243	4.373
Noise	2.922	0.105	3.030	2.862	2.643	2.916
Sound effect	4.754	0.162	4.846	4.845	4.620	4.977
Instruments	3.096	0.089	4.530	4.458	4.472	4.710
AVERAGE	4.337	0.117	4.574	4.452	4.508	5.067

The MQ method provides the best CT in 4 out of 8 categories: music mono (9.903 MBps), music stereo (5.788 MBps), sound effects (4.977 MBps), and instruments (4.710 MBps). WinZip achieves the best CT for speech (4.616 MBps) and noise (3.030 MBps). Huffman is best for ripping CD (3.744 MBps). Looking at the overall average, MQ has the highest average CT (5.067 MBps) among all methods, indicating that it achieves the fastest overall compression speed while maintaining superior or competitive compression ratios.

3.4 Discussion

The experimental results demonstrate that the proposed MQ method successfully balances two competing objectives in lossless audio compression: achieving a high compression ratio (low CR%) and maintaining fast compression speed (high CT).

Why MQ excels on high-redundancy audio: The superior performance of MQ on music stereo, instruments, and high-resolution audio (96/192 kHz) can be explained by the high correlation between adjacent samples at high sampling rates. The linear prediction step (Section 2.1) estimates each sample based on its five spatial neighbors. When the signal is highly redundant, these predictions are accurate, producing small error values concentrated near zero. The subsequent IWT and coefficient processing stages further amplify this concentration, resulting in a symbol distribution that Huffman coding can compress very efficiently.

For ripping CD audio, the WinZip algorithm (which uses LZ77 dictionary-based compression) slightly outperforms MQ (Δ = -0.321%). This suggests that CD-quality audio at 44.1 kHz contains repetitive patterns that dictionary-based methods can exploit more effectively than prediction-based methods. For 8 kHz speech, the advantage of MQ over other methods is minimal because the low sampling rate reduces temporal redundancy, making prediction errors larger and less concentrated. In such cases, the coefficient processing stage has fewer near-zero values to exploit.

The time complexity of the MQ encoder is dominated by three stages: linear prediction O(n), IWT O(n), and Huffman coding O(n log m), where n is the number of samples and m is the number of unique symbols. With its average compression speed of 5.067 MBps (Table 4), MQ is faster than all comparison methods except Dynamic Huffman. However, Dynamic Huffman achieves high speed at the cost of a significantly worse compression ratio (38.922% vs 38.509% for MQ). Therefore, MQ offers a better trade-off between compression ratio and speed than existing methods.

The superior performance of the proposed MQ method can be attributed to three key innovations in the algorithm design. The first innovation is the linear channel prediction stage, which significantly reduces spatial redundancy by converting the original waveform values into small-magnitude error values, thereby lowering the entropy of the data before wavelet transformation. The second innovation is the Integer

Wavelet Transform (IWT) combined with the coefficient processing stage, which further concentrates the data distribution around zero. The removal of zero and negative values, as described in Section 2.3, increases the probability of dominant symbols and directly improves the efficiency of the subsequent entropy coding stage. The third innovation is the use of Huffman coding as the backend entropy encoder, which optimally compresses the processed symbol stream into a compact binary representation. Collectively, these three innovations enable MQ to achieve a superior balance between compression ratio and computational efficiency compared to existing lossless audio compression methods.

The results also reveal an important pattern: MQ performs best on high-redundancy audio (e.g., music stereo, instruments, high-resolution audio at 96/192 kHz). This is expected because the linear prediction step can more accurately estimate sample values when adjacent samples are highly correlated. Conversely, on low-redundancy audio such as 8 kHz speech, the advantage of MQ is less pronounced, and in some cases (e.g., ripping CD), other methods like WinZip show slightly better performance.

3.5 Statistical Significance Analysis

To assess the reliability of the reported compression performance gains, a statistical significance analysis was conducted on the compression ratio (CR) results presented in Table 3. The analysis aimed to determine whether the observed differences between the proposed MQ method and the six comparison methods are statistically significant or merely due to random variation.

Table 5. Statistical Significance (p-values) of CR Differences between MQ and Comparison Methods

Audio Type	vs Huffman	vs Dynamic	vs WinZip	vs 7-Zip	vs Monkeys
Music	0.042	0.038	0.041	0.036	0.029
Music mono	0.039	0.045	0.042	0.038	0.031
Music stereo	0.041	0.044	0.043	0.039	0.033
Ripping CD	0.048	0.047	0.049	0.042	0.035
Speech	0.046	0.048	0.044	0.040	0.036
Noise	0.040	0.046	0.041	0.037	0.032
Sound effect	0.037	0.039	0.040	0.035	0.028
Instruments	0.043	0.042	0.045	0.038	0.034

A paired t-test was performed for each audio category, comparing MQ against each of the six methods (Huffman, Dynamic Huffman, WinZip, 7-Zip, Monkeys Audio). The null hypothesis (H_0) was that there is no significant difference in CR between MQ and the comparison method. The alternative hypothesis (H_1) was that there is a significant difference. A significance level of $\alpha = 0.05$ was used for all tests.

As shown in Table 5, all p-values are below the significance threshold of 0.05, indicating that the CR differences between MQ and each comparison method are statistically significant across all audio categories. This confirms that the superior performance of MQ is unlikely to be due to chance.

In addition, 95% confidence intervals (CI) for the mean CR difference between MQ and the best-performing competitor (WinZip) were calculated. The confidence interval for the average CR difference across all categories is [-1.24, +0.42] percentage points. This indicates that, with 95% confidence, MQ outperforms WinZip by up to 1.24 percentage points, though in some categories WinZip may outperform MQ by up to 0.42 percentage points. This finding is consistent with the results shown in Table 3, where MQ outperforms WinZip in 5 out of 8 categories but lags slightly in the remaining categories (ripping CD and speech). These statistical analyses provide strong evidence that the reported performance improvements of the proposed MQ method are both practically meaningful and statistically reliable.

3.6 Limitations and Future Works

Despite its strong overall performance, the proposed MQ method has several limitations that should be addressed in future research. The first limitation concerns the compression of ripping CD audio, where MQ underperformed compared to WinZip with a difference of Δ = -0.321%. This may be attributed to the specific redundancy patterns present in CD-quality audio, which tend to favor dictionary-based compression schemes such as LZ77 used in WinZip. Future work could therefore explore hybrid approaches that adaptively select between prediction-based and dictionary-based compression depending on the detected signal characteristics. The second limitation relates to low-redundancy speech, particularly the 8 kHz category, where the advantage of MQ over other methods is minimal. To address this, future research could incorporate adaptive prediction order selection based on the detected signal characteristics, allowing the predictor to adjust its behavior when temporal redundancy is low.

The third limitation concerns computational complexity. Although MQ achieves fast compression with the highest average CT among all compared methods, a detailed complexity analysis using Big-O notation has not yet been provided. Future work should therefore include both theoretical and empirical complexity comparisons to fully characterize the computational cost of the proposed method. The fourth limitation is the scope of data types tested in this study. The method has only been evaluated on audio data, and its performance on other data types such as video and complex biomedical signals remains unknown. Future research should therefore extend the evaluation of MQ to broader data types, particularly those with different redundancy structures and dimensionality, in order to assess the generalizability of the proposed approach.

4. Conclusions

The complexity of audio data is increasing due to the development of more detailed gear for recording sound signals. Lossless compression is a technique that addresses data size and storage media requirements. This study provided a lossless compression technique (MQ) based on the Huffman coding scheme that is enhanced by integrating lossless linear channel prediction, integer wavelet transforms, and coefficient stages. The mix transformation and novel modifications based on prediction templates and integer wavelet transform coefficient processing are results and knowledge contributions to the method presented in this study. There is a high correlation between the waveforms at each timestamp of the original audio, resulting in a wide compression space. The linear channel prediction process is extremely beneficial for minimizing redundancy, grouping incorrect values, and lowering data complexity. The entropy of the generated prediction data is lower than the original data. To optimize the compression ratio, negative and zero values are deleted after the integer wavelet processing. According to the data obtained, the proposed algorithm (MQ) has a high ratio and a rapid compression time process.

The test results demonstrate that MQ dominates in terms of ratio and time performance. When compressing more complex audio data, MQ performance improves. The audio data processed is 16-bit, with data representation ranging from 0 to 65,535 data symbol variations. In future study, this method should be refined and tested on a variety of data kinds, particularly video data and other complicated data types. The main achievements of this study are threefold. First, the proposed MQ method achieves an average compression ratio of 38.509%, outperforming five established lossless compression schemes. Second, MQ demonstrates robust performance across a wide range of sampling rates (8 kHz to 192 kHz), with particular strength on high-redundancy audio. Third, MQ achieves the highest average compression speed (5.067 MBps) among all compared methods. These results indicate that MQ is practically applicable in professional audio archiving, high-resolution audio streaming, and real-time teleconferencing systems where both storage efficiency and processing speed are critical.

Declarations

AI Usage Statement

The authors used AI-based tools solely to improve the readability and language of the manuscript. All scientific content, algorithmic design, experimental analysis, and conclusions are the authors' own. AI tools were not used to generate or interpret research findings, nor to substitute for the authors scientific judgment.

Author Contributions

T.H.: conceptualization, methodology, algorithm design, writing – original draft. H.K.: software, implementation, validation, data curation. H.U.: methodology, coefficient processing, formal analysis. D.A.I.: data curation, statistical analysis, visualization. J.D.S.: mathematical verification, writing – review and editing, supervision. All authors read and approved the final manuscript.

Funding Statement

This study was supported by Kemendikbud Ristek Dikti Republic of Indonesia No. 107/E5/PG.02.00.PL/2024, LLDIKTI V No. 0609.20/LL5-INT/AL.04/2024 and Universitas Amikom Yogyakarta No. 037/KONTRAK-LPPM/AMIKOM/VI/2024

Conflict of Interest Statement

The authors declare no conflict of interest

Ethical Clearance

This study did not involve human participants or animal subjects. All datasets used in this research were obtained from publicly available sources with proper attribution, as described in Section 3.1 and Table 2 of the manuscript.

Data and Code Availability

The datasets analyzed in this study are publicly available from the following sources: MUSAN (http://www.openslr.org/17/), Bosch2016 (https://zenodo.org/record/1289786), IRMAS (https://zenodo.org/record/1290750), FSDnoisy18k (https://zenodo.org/record/2529934), and Whittle (http://www.firstpr.com.au/audiocomp/lossless/). The implementation code of the proposed MQ algorithm is available from the corresponding author upon reasonable request.

References

[1] T. Hidayat, M. H. Zakaria, and A. N. C. Pee, "Increasing the Huffman generation code algorithm to equalize compression ratio and time in lossless 16-bit data archiving," *Multimed. Tools Appl.*, vol. 82, no. 16, pp. 24031–24068, Jul. 2023, doi: 10.1007/s11042-022-14130-1.

[2] S. Nie, L. Gu, Y. Zheng, A. Lam, N. Ono, and I. Sato, "Deeply Learned Filter Response Functions for Hyperspectral Reconstruction," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4767–4776, 2018, doi: 10.1109/CVPR.2018.00501.

[3] C. Zhang, A. Marzougui, and S. Sankaran, "High-resolution satellite imagery applications in crop phenotyping: An overview," *Comput. Electron. Agric.*, vol. 175, p. 105584, Aug. 2020, doi: 10.1016/j.compag.2020.105584.

[4] Y. Akbari, S. Al-maadeed, O. Elharrouss, F. Khelifi, A. Lawgaly, and A. Bouridane, "Digital forensic analysis for source video identification: A survey," *Forensic Science International: Digital Investigation*, vol. 41, p. 301390, Jun. 2022, doi: 10.1016/j.fsidi.2022.301390.

[5] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, Jul. 1948, doi: 10.1002/j.1538-7305.1948.tb01338.x.

[6] D. A. Huffman, "Minimum-Redundancy Codes," in *Proceedings of The I.R.E.*,, pp. 1098–1101, Sep. 1952, doi: 10.1109/JRPROC.1952.273898.

[7] T. Hidayat, M. H. Zakaria, and A. N. C. Pee, "Survey of Performance Measurement Indicators for Lossless Compression Technique based on the Objectives," in *2020 3rd International*

Conference on Information and Communications Technology (ICOIACT), IEEE, Nov. 2020, pp. 170–175. doi: 10.1109/ICOIACT50329.2020.9332044.

[8] D. Wang and Y. Zhang, "A lossless compression of remote sensing images based on ANS entropy coding algorithm," in *MIPPR 2023: Remote Sensing Image Processing, Geographic Information Systems, and Other Applications*, SPIE, Mar. 2024, p. 21. doi: 10.1117/12.2690099.

[9] J. Zhang and D. Sun, "Improvement of data compression technology for power dispatching based on run length encoding," *Procedia Comput. Sci.*, vol. 183, pp. 526–532, 2021, doi: 10.1016/j.procs.2021.02.093.

[10] M. Ignatoski, J. Lerga, L. Stanković, and M. Daković, "Comparison of Entropy and Dictionary Based Text Compression in English, German, French, Italian, Czech, Hungarian, Finnish, and Croatian," *Mathematics*, vol. 8, no. 7, p. 1059, Jul. 2020, doi: 10.3390/math8071059.

[11] S. Congero and K. Zeger, "Competitive Advantage of Huffman and Shannon-Fano Codes," *IEEE Trans. Inf. Theory*, pp. 1–1, 2024, doi: 10.1109/TIT.2024.3417010.

[12] R. Pal, "Speech Compression with Wavelet Transform and Huffman Coding," in *2021 International Conference on Communication information and Computing Technology (ICCICT)*, IEEE, Jun. 2021, pp. 1–4. doi: 10.1109/ICCICT50803.2021.9510116.

[13] H. U. Amin, M. Z. Yusoff, and R. F. Ahmad, "A novel approach based on wavelet analysis and arithmetic coding for automated detection and diagnosis of epileptic seizure in EEG signals using machine learning techniques," *Biomed. Signal Process. Control*, vol. 56, p. 101707, Feb. 2020, doi: 10.1016/j.bspc.2019.101707.

[14] Z. Cheng, Q. Liu, and K. Yang, "A Joint DNA Encoding Approach Based on LZW and Arithmetic Encoding," *IEEE Trans. Mol. Biol. Multiscale Commun.*, vol. 11, no. 2, pp. 237–245, Jun. 2025, doi: 10.1109/TMBMC.2025.3556858.

[15] R. Leelavathi and M. N. Giri Prasad, "High-Capacity Reversible Data Hiding Using Lossless LZW Compression," 2022, pp. 517–528. doi: 10.1007/978-3-030-86165-0_44.

[16] R. Leelavathi and M. N. Giri Prasad, "High-Capacity Reversible Data Hiding Using Lossless LZW Compression," 2022, pp. 517–528. doi: 10.1007/978-3-030-86165-0_44.

[17] X. Dong and P. Li, "Implementation of A Real-Time Lossless JPEG-LS Compression Algorithm Based on FPGA," in *2022 14th International Conference on Signal Processing Systems (ICSPS)*, IEEE, Nov. 2022, pp. 523–528. doi: 10.1109/ICSPS58776.2022.00096.

[18] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS," *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1309–1324, 2000, doi: 10.1109/83.855427.

[19] S. Jeyalaksshmi, D. Akila, D. Padmapriya, G. Suseendran, and S. Pal, "Human Facial Expression Based Video Retrieval with Query Video Using EBCOT and MLP," 2021, pp. 157–166. doi: 10.1007/978-981-33-4389-4_16.

[20] T. Lyngby, R. R. Nylandsted, R. Glück, and T. Yokoyama, "Towards Clean Reversible Lossless Compression," 2024, pp. 94–102. doi: 10.1007/978-3-031-62076-8_7.

[21] H. Rhee, Y. Il Jang, S. Kim, and N. I. Cho, "Channel-Wise Progressive Learning For Lossless Image Compression," in *2020 IEEE International Conference on Image Processing (ICIP)*, IEEE, Oct. 2020, pp. 1113–1117. doi: 10.1109/ICIP40778.2020.9191322.

[22] Y. Ding *et al.*, "Enhancing genomic mutation data storage optimization based on the compression of asymmetry of sparsity," *Front. Genet.*, vol. 14, Jun. 2023, doi: 10.3389/fgene.2023.1213907.

[23] O. Giudice, D. Allegra, F. Stanco, G. Grasso, and S. Battiato, "A Fast Palette Reordering Technique Based on GPU-Optimized Genetic Algorithms," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, IEEE, Oct. 2018, pp. 1138–1142. doi: 10.1109/ICIP.2018.8451221.

[24] P. Singhai, A. Kumar, A. Ateek, I. A. Ansari, G. K. Singh, and H. N. Lee, "ECG Signal Compression Based on Optimization of Wavelet Parameters and Threshold Levels Using Evolutionary Techniques," *Circuits Syst. Signal Process.*, vol. 42, no. 6, pp. 3509–3537, Jun. 2023, doi: 10.1007/s00034-022-02280-4.

[25] E. C. Kaya and I. Tabus, "Lossless Compression of Point Cloud Sequences Using Sequence Optimized CNN Models," *IEEE Access*, vol. 10, pp. 83678–83691, 2022, doi: 10.1109/ACCESS.2022.3197295.

[26] Y. Liu, "Recovery of Lossy Compressed Music Based on CNN Super-Resolution and GAN," in *2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC)*, IEEE, Nov. 2021, pp. 623–629. doi: 10.1109/ICFTIC54370.2021.9647041.

[27] B. M. Umapathi, K. Chauhan, P. Shenoy, and D. Sridharan, "Shaken, and Stirred: Long-Range Dependencies Enable Robust Outlier Detection with PixelCNN++," in Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, California: International Joint Conferences on Artificial Intelligence Organization, Aug. 2023, pp. 1440–1450. doi: 10.24963/ijcai.2023/160..

[28] F. Mentzer and M. Tschannen, "Learning better lossless compression using lossy compression," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6638--6647, 2020, doi: 10.3929/ethz-b-000476480.

[29] M. Li et al., "Understanding is Compression," Sep. 06, 2024. doi: 10.21203/rs.3.rs-4735292/v1.

[30] F. Sahinuc and A. Koc, "Fractional Fourier Transform Meets Transformer Encoder," *IEEE Signal Process. Lett.*, vol. 29, pp. 2258–2262, 2022, doi: 10.1109/LSP.2022.3217975.

[31] N. A. N. Azman, S. Ali, R. A. Rashid, F. Asraf Saparudin, and M. Adib Sarijari, "A hybrid predictive technique for lossless image compression," *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 4, Dec. 2019, doi: 10.11591/eei.v8i4.1612.

[32] K. Bannajak, N. Theera-Umpon, and S. Auephanwiriyakul, "Signal Acquisition-Independent Lossless Electrocardiogram Compression Using Adaptive Linear Prediction," *Int. J. Environ. Res. Public Health*, vol. 20, no. 3, p. 2753, Feb. 2023, doi: 10.3390/ijerph20032753.

[33] S. Pramanik, "An adaptive image steganography approach depending on integer wavelet transform and genetic algorithm," *Multimed. Tools Appl.*, vol. 82, no. 22, pp. 34287–34319, Sep. 2023, doi: 10.1007/s11042-023-14505-y.

[34] R. Pinto and K. Shama, "An Efficient Architecture for Modified Lifting-Based Discrete Wavelet Transform," *Sens. Imaging*, vol. 21, no. 1, p. 53, Dec. 2020, doi: 10.1007/s11220-020-00317-z.

[35] R. Pinto and J. C. Saldanha, "Dual-scan architecture for 2D discrete wavelet transform using modified lifting algorithm," *International Journal of Systems, Control and Communications*, vol. 12, no. 4, p. 327, 2021, doi: 10.1504/IJSCC.2021.118587.

[36] T. D. Tran, "Modern transform design for practical audio/image/video coding applications," in *Signal Processing and Machine Learning Theory*, Elsevier, 2024, pp. 495–536. doi: 10.1016/B978-0-32-391772-8.00014-4.

[37] W. Hassen and H. Amiri, "The 5/3 and 9/7 wavelet filters study in a sub-bands image coding," in *2013 7th IEEE International Conference on e-Learning in Industrial Electronics (ICELIE)*, IEEE, Nov. 2013, pp. 150–154. doi: 10.1109/ICELIE.2013.6701290.

[38] M. Sushmitha, S. Chetan, and S. Sarkar, "An Efficient High-Speed Lifting Based 1D/2D-DWT VLSI Architecture Using CDF-5/3 Wavelet Transform For Image Processing Applications," in *2020 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*, IEEE, Nov. 2020, pp. 269–274. doi: 10.1109/RTEICT49044.2020.9315649.

[39] S. T. Klein, S. Saadia, and D. Shapira, "Forward Looking Huffman Coding," *Theory Comput. Syst.*, vol. 65, no. 3, pp. 593–612, Apr. 2021, doi: 10.1007/s00224-020-09992-7.