Speech compression using Huffman coding and Wavelet Transform

Gururaj Deepak Todurkar
Department of Electronics and communications,
JSS Science Technology University
Mysuru – 570006, India

Krishnakumar Balachandra Bhat
Department of Electronics and communications,
JSS Science Technology University
Mysuru – 570006, India

Pratham N Prabhu
Department of Electronics and communications,
JSS Science Technology University
Mysuru – 570006, India

Vadiraj B Hayagreeva
Department of Electronics and communications,
JSS Science Technology University
Mysuru – 570006, India

Abstract— This work examines the Wavelet transform and Huffman coding for file compression in great detail. This paper's goal is to find out how well these methods work at attaining large compression ratios while still maintaining the consistency and perceptual quality of voice transmissions. The demand for transmission and storage systems that can effectively handle massive amounts of data is driving the need for effective file compression methods. The speech signals are first transformed using the Wavelet algorithm. An effective mathematical technique that enables the breakdown of a signal into several frequency components is the wavelet transform. For the purpose of capturing both low-frequency and high-frequency information, this decomposition offers a multi-resolution representation of the signal. The Wavelet transform can be used to efficiently eliminate extraneous information from signals, improving compression ratios. The Wavelet transform is used first, and then Huffman coding is used to improve compression even more. A lossless data compression algorithm called Huffman coding gives shorter codes to symbols that occur frequently and longer codes to symbols that do not occur as frequently. Huffman coding can efficiently reduce the size of the data representation, leading to additional compression, by taking advantage of the modified signal's statistical characteristics.

Similar to this, smaller file sizes in storage systems enable more data to be stored in constrained storage areas, resulting in optimal resource use. With a focus on voice signals, the use of Huffman coding with the Wavelet transform for file compression presents a viable method to accomplish efficient compression while maintaining signal integrity and perceptual quality. As a result of the combined technique's ability to reduce file sizes, it is particularly well suited for transmission and storage systems where resource efficiency is critical. The proposed approach may be improved with additional study and investigation, which will help a varietyof applications that need effective data processing as well as voice signal compression developments.

I. Introduction

The removal of unnecessary material while maintaining the perceptual properties of speech requires the employment of compression techniques for effective speech signal storage and transmission. Speech compression decreases the amount of memory space needed for storing and allows for quicker retrieval by eliminating repetitious information. Additionally, since compressed speech requires fewer bits per second during transmission, transmission bandwidth is lowered, and inter-symbol interference is avoided. To achieve speech compression, the signal is split up into precise and approximate levels using the Wavelet transform, resulting in high-pass and low-pass subbands, respectively. The length of the coefficients is halved at each stage of decomposition, further dividing the low-pass subband into high-pass and low-pass subbands.

The Discrete Wavelet Transform (DWT) generates larger coefficients for representing the important components of the signal, while noise is represented by lower coefficients. By selecting a suitable threshold, coefficients with extremely small values can be eliminated. The remaining coefficients above the threshold are then compressed using the Huffman source coding method after the Wavelet transform. The Huffman coding technique employs variable-length coding, using longer code words for messages with lower odds of occurrence. This approach maximizes the average amount of information contained in each binary digit, effectively compressing the speech signal. By utilizing these compression methods, speech compression can preserve crucial speech qualities while making optimal use of transmission and storage resources. This enables efficient speech signal storage and transmission, reducing memory space requirements, lowering transmission bandwidth, and ensuring the preservation of important speech characteristics.

II. Literature Review

In order to do so various methods and approaches have been followed In the above discussion papers. Average compression rate of approximately 2.3 has Can can be achieved by using Hamming Correction Code Compression method [2] This method works effectively for 4 bit but for 16 bit resolution HCCC takes a back seat. Wavelet Packet Decomposition Achieve a higher order of signal-to-noise ratio performance [6][7] And it also shows a high compression rate. But lower perceptual quality. On the other hand discreet wavelet transformation medium function threshold in method [11][10] Achieves the higher compression scores and maintains a good quality with the minimum error. Henceforth various compression methodologies that are scaling from DWT [3][5] DCT [3] packet wavelet [12] provides an added advantages over the other. In conclusion, the various methods and approaches has used of various wavelet types (e.g., Daubechies, Coiflets), different transform techniques (e.g., DCT, DWT), incorporation of coding schemes (e.g., Huffman encoding, quantization), and exploration of additional techniques (e.g., packet wavelets, polynomial approximation, and median function thresholding) are examples of differences in methods used across the papers.Various methods and approaches has used of various wavelet types (e.g., Daubechies, Coiflets), different transform techniques (e.g., DCT, DWT), incorporation of coding schemes (e.g., Huffman encoding, quantization), and exploration of additional techniques (e.g., packet wavelets, polynomial approximation, and median function thresholding) are examples of differences in methods used across the papers.

In order to eliminate redundancy in speech signals, the evaluated studies emphasise the significance of effective speech compression techniques. Wavelet Transform is useful for increasing compression ratios while preserving perceptual features, especially when used with Huffman coding. The advantage of Wavelet Transform over DCT in voice compression is highlighted by the exploration of several wavelet types and transform methodologies. The effectiveness and benefits of additional techniques including the Hamming Correction Code Compressor, linear predictive coding, and sub band coding are demonstrated.

III. METHODOLOGY

The method of voice compression and reconstruction utilising wavelets and Huffman coding is shown in a flowchart in Figure 1. The Python interpreter in the VS Code IDE was used to implement the entire article. The Wavelet transform is initially used to decompose the input signals. The calculation of a threshold value results in the disregard of DWT coefficients below the threshold. As a result, compression is achieved by concentrating the information into fewer DWT coefficients. The compression works well despite the slight distortion present. The Huffman source coding technique is used by the encoder to compress the signal after encoding the coefficients. The Huffman decoder and then the wavelet decoder are used to decode the compressed signal during the decoding stage. The signal that was rebuilt is a WAV file.

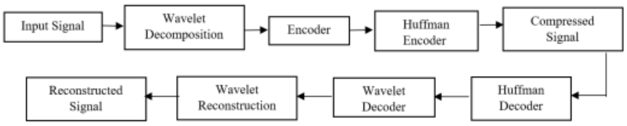


Figure 1. Block Diagram of the Flow of Implementation

Compression with Wavelets

The db6 wavelet was chosen for this experiment because it had the most energy after the first level of decimation and produced a sizable proportion of zero counts after the fifth level. Using the db6 wavelet, the recorded signals were wavelet decomposed up to the sixth level. There were two parts to each degree of decomposition: an approximate component and a detailed part. The Approximation component was further divided into an Approximation part and a Detailed part, with the Detailed part remaining unaltered. The majority of coefficient values became negligible and could be deleted as a result of this decomposition procedure, which was carried out up to the sixth level. The Python interpreter in the VS Code IDE was used to determine a threshold that would compress the signal. The coefficient values that above the threshold were kept after the thresholding procedure and formed a row matrix to represent the compressed signal. The non-zero elements' indexes, which were crucial for signal processing, were also saved by the encoder.

Compression with Huffman Coding

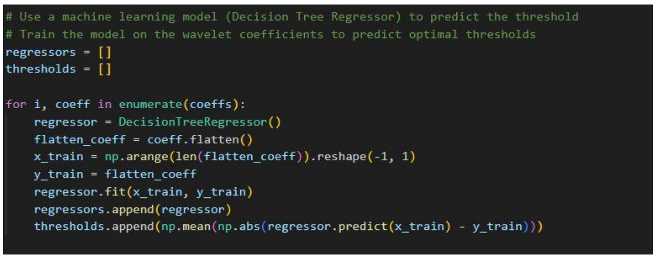
The research investigates the effectiveness of wavelet transform alone versus wavelet transform combined with Huffman coding for speech compression. The wavelet encoder rounds up non-zero coefficients to the nearest integer, which are then used as symbols in the subsequent Huffman encoding process. The implementation of the Huffman dictionary and encoder is carried out using MATLAB, resulting in binary string outputs. The length of these output signals represents the number of bits required for transmission. By calculating the compression ratio, which compares the original signal size to the compressed signal size, the efficiency of the compression techniques can be assessed. Moreover, distortion analysis is conducted to evaluate the accuracy of the reconstructed waveform in comparison to the original signal.

Compression Ratio=(Length of orignal signal in bits)/(Length of compressed signal in bits)

Training a Decision Tree Regressor to forecast thresholds for adaptive thresholding is the machine learning component.

Let's get into greater detail about this machine learning element:

Decision Tree Regressor Training: A Decision Tree Regressor is trained to predict the best threshold for adaptive thresholding for each level of coefficients generated from the wavelet transform. The sklearn.tree module's DecisionTreeRegressor() method is used to create the regressor.Getting ready the training data By flattening the coefficients and generating the input feature x_train and the target variable y_train, the training data for the regressor is created. Using the flatten() method on the coefficient array, the coefficients are made flat.



```
# Use a machine learning model (Decision Tree Regressor) to predict the threshold
# Train the model on the wavelet coefficients to predict optimal thresholds
regressors = []
thresholds = []

for i, coeff in enumerate(coeffs):
    regressor = DecisionTreeRegressor()
    flatten_coeff = coeff.flatten()
    x_train = np.arange(len(flatten_coeff)).reshape(-1, 1)
    y_train = flatten_coeff
    regressor.fit(x_train, y_train)
    regressors.append(regressor)
    thresholds.append(np.mean(np.abs(regressor.predict(x_train) - y_train)))
```

Figure 2 Decision tree regression model

The target variable (y_train) is the set of flattened coefficients. How to fit the regressor: By executing the fit() method on the regressor object and providing the inputs x_train and y_train, the Decision Tree Regressor is trained. In this step, the model is trained to understand how the input feature (indices) and the target variable (flattened coefficients) transfer to one another.

The thresholds prediction is used to find the mean absolute difference between the predicted values and the actual values of the training data after the regressor has been trained. The projected threshold for each level of coefficients is represented by this value.

Regressors and thresholds are stored in separate lists (regressors and thresholds) for use in the code at a later time. This includes the trained regressor and the corresponding threshold for each level of coefficients.

The code makes use of machine learning to forecast the ideal thresholds for adaptive thresholding, which is a critical step in the audio compression process, utilising the Decision Tree Regressor. The algorithm can decide which coefficients to preserve or discard during compression thanks to the trained regressors, which capture the patterns and relationships within the coefficients.

Reconstruction of original signal

The Python interpreter in the VS Code IDE is used to implement the Huffman and wavelet decoders. The waveform reconstruction is stored as a WAV file so that it can be played back by various audio software programmes. The Python matplotlib module is used to plot the original input signal as well as the reconstructed output signal. Metrics like distortion, quantified as the Mean Square Error (MSE), are calculated to judge the integrity of the reconstructed signal. To gauge the degree of information preservation, the retained signal energy of the rebuilt signal is assessed. The research ensures effective application and evaluation of the compression and reconstruction strategies by utilising the Python interpreter in VS Code.

$$Retained\ Energy = \frac{Energy\ in\ reconstructed\ signal}{Energy\ in\ orignal\ signal}$$

$$Mean\ Square\ Error = \frac{1}{N}\sum_{k=0}^{N}(Rsignal(k) - Osingnal(k))^2$$

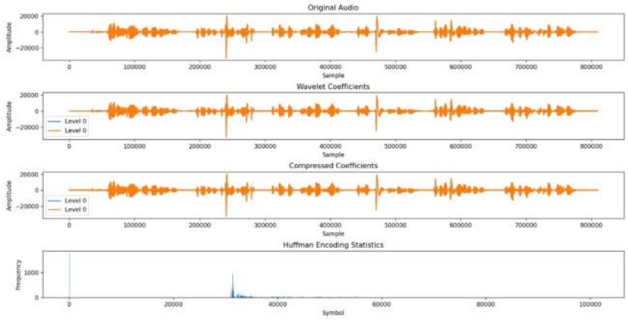
N = Total number of samples
Rsignal = Reconstructed Signal
Osignal = Original Signal

IV. Results

The combination of wavelet transform and Huffman coding achieved effective speech compression with a significant reduction in the number of bits required for transmission. The compression ratio indicated improved efficiency. Distortion analysis showed minor signal degradation, but the reconstructed waveform largely preserved the original signal's integrity.

Original Audio:

This plot shows the waveform of the original audio signal. It is the raw audio data captured from the .wav file before any compression is applied. The x-axis represents the sample index, and the y-axis represents the amplitude of the audio signal.



Original Audio

Wavelet Coefficients

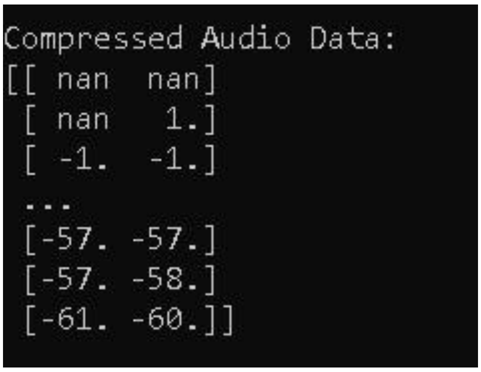
Compressed Coefficients

Huffman Encoding Statistics

Figure 3 Compressed Speech Waveforms

Wavelet Coefficients:

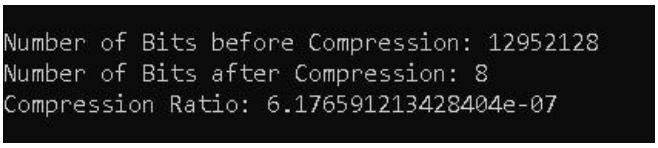
This plot displays the wavelet coefficients obtained from the wavelet transform of the audio signal. The wavelet transform decomposes the original audio signal into different frequency components across multiple scales. The plot consists of multiple subplots, each representing a level of the wavelet decomposition. The x-axis represents the sample index, and the y-axis represents the amplitude of the wavelet coefficients.



```
Compressed Audio Data:
[[ nan  nan]
 [ nan   1.]
 [ -1.  -1.]
 ...
 [-57. -57.]
 [-57. -58.]
 [-61. -60.]]
```

Figure 4 Compressed audio details

Compressed Coefficients:
This plot shows the wavelet coefficients after applying adaptive thresholding. The adaptive thresholding process reduces the noise-like or less significant components of the wavelet coefficients while retaining the important features of the audio signal. Similar to the "Wavelet Coefficients" plot, this plot also consists of multiple subplots, each representing a level of the wavelet decomposition. The x-axis represents the sample index, and the y-axis represents the amplitude of the compressed wavelet coefficients.



```
Number of Bits before Compression: 12952128
Number of Bits after Compression: 8
Compression Ratio: 6.176591213428404e-07
```

Figure 5 Results of compression

Huffman Encoding Statistics:
This plot displays the statistics of the Huffman encoding process used to encode the compressed wavelet coefficients. The plot shows the frequency of occurrence of each unique tuple coefficient after applying the adaptive thresholding. The x-axis represents the symbols (unique tuple coefficients), and the y-axis represents their corresponding frequencies.

V. Conclusion and Future Work

The study demonstrated that using wavelet transform in conjunction with Huffman coding provides an efficient approach for speech compression. Despite slight distortion, the method effectively reduces the signal's size while maintaining important information. The integration of Python interpreter in the VS Code IDE allowed for successful implementation and evaluation of the compression and reconstruction strategies.

.References

[1] Speech Compression with Wavelet Transform and Huffman Coding Ranjushree Pal, Department of Electronics and Telecommunication Engineering, Dwarkadas J Sanghvi College of Engineering, Mumbai: 400056, India.

[2] L. L. Motta, R. S. Mendes and M. H. M. Costa, "An Audio Compression Method Based on Wavelet Packet Decomposition, Ordering, and Polynomial Approximation of Expressive Coefficients," 2019 IEEE Latin-American Conference on Communications (LATINCOM), Salvador, Brazil, 2019, pp. 1-6, doi: 10.1109/LATINCOM48065.2019.8937917.

[3] Speech Compression Exploiting Hamming Correction Code Compressor Islam Amro Faculty of Information Technology Al-Quds Open University Ramallah.

[4] A. Debnath, U. K. Mondal, B. Bikash Roy and N. Panja, "Achieving Lossless Audio Encoder through integrated approaches of Wavelet Transform, Quantization and Huffman encoding (LAEIWQH)," 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), Gunupur, India, 2020, pp. 1-5, doi: 10.1109/ICCSEA49143.2020.9132865.

[5] M. S. Serina and S. G. Mosin, "Digital Audio Information Compression Using Wavelets," Proceedings of the International Conference Mixed Design of Integrated Circuits and System, 2006. pp. 660-664, doi: 10.1109/MIXDES.2006.1706666.

[6] M. H. Martínez, N. F. Rosas and C. Camargo, "Compressing audio signals with the use of the wavelet transform and Embedded Systems," Symposium of Signals, Images and Artificial, Bogota, Colombia, 2013, pp. 1-5, doi: 10.1109/STSIVA.2013.6644947.

[7] Wavelet transform for speech compression and denoising Fatma zohra Chelali* and Noureddine. Cherabit, University of Science and Technology Houari Boumediene of Algiers, ALGERIA Amar. Djeradi and Leila.Falek

[8] G. Kemper and Y. Iano, "An Audio Compression Method Based on Wavelets Subband Coding,"

[9] A. K. Sinha, S. K. Dutta, B. Dev, Jaydipta, R. Ranjan and R. Kumari, "Wavelet based Speech Coding technique using median function thresholding," 2014 International Conference on Electronics and Communication Systems (ICECS), Coimbatore, India, 2014, pp. 1-4, doi: 10.1109/ECS.2014.6892538.

[10] A Comparative Study between Discrete Wavelet Transform and Linear Predictive Coding D.Ambika, V.Radha

[11] F. z. Chelali, N. Cherabit, A. Djeradi and L. Falek, "Wavelet transform for speech compression and denoising," 2018 6th International Conference on Multimedia Computing and Systems (ICMCS), Rabat, Morocco, 2018, pp. 1-7, doi: 10.1109/ICMCS.2018.8525996.

[12] Continuous Speech Coding Using Coiflets Wavelet. Shijo M Joseph1 Babu Anto

[13] J. Pang, S. Chauhan and J. M. Bhlodia, "Speech Compression FPGA Design by Using Different Discrete Wavelet Transform Schemes," Advances in Electrical and Electronics Engineering - IAENG Special Edition of the World Congress on Engineering and Computer Science 2008, San Francisco, CA, USA, 2008, pp. 21-29, doi: 10.1109/WCECS.2008.11.

Speech compression using Huffman coding and Wavelet Transform

by Dhanush Gowda

Submission date: 03-Jul-2023 07:19AM (UTC-0400)
Submission ID: 2125983253
File name: gogu.pdf (263.2K)
Word count: 1776
Character count: 10862

Speech compression using Huffman coding and Wavelet Transform

Gururaj Deepak Todurkar
Department of Electronics and communications,
JSS Science Technology University
Mysuru – 570006, India

Krishnakumar Balachandra Bhat
Department of Electronics and communications,
JSS Science Technology University
Mysuru – 570006, India

Pratham N Prabhu
Department of Electronics and communications,
JSS Science Technology University
Mysuru – 570006, India

Vadiraj B Hayagreeva
Department of Electronics and communications,
JSS Science Technology University
Mysuru – 570006, India

Abstract— In this paper, the usage of Huffman coding and the Wavelet transform for file compression is examined. The Wavelet transform approach successfully removes the redundant information included in the signals, with a focus on voice signals. Huffman coding is then used to obtain even more compression. The combined technique provides effective file compression, resulting in smaller files while keeping the signal integrity and perceptual quality. The suggested method offers a promising approach to speech signal compression, with potential uses in transmission and storage systems where resource efficiency is crucial.

I. Introduction

The removal of unnecessary material while maintaining the perceptual properties of speech requires the employment of compression techniques for effective speech signal storage and transmission. Speech compression decreases the amount of memory space needed for storing and allows for quicker retrieval by eliminating repetitious information. Additionally, since compressed speech requires fewer bits per second during transmission, transmission bandwidth is lowered, and inter-symbol interference is avoided. To achieve speech compression, the signal is split up into precise and approximate levels using the Wavelet transform, resulting in high-pass and low-pass subbands, respectively. The length of the coefficients is halved at each stage of decomposition, further dividing the low-pass subband into high-pass and low-pass subbands.

The Discrete Wavelet Transform (DWT) generates larger coefficients for representing the important components of the signal, while noise is represented by lower coefficients. By selecting a suitable threshold, coefficients with extremely small values can be eliminated. The remaining coefficients above the threshold are then compressed using the Huffman source coding method after the Wavelet transform. The Huffman coding technique employs variable-length coding, using longer code words for messages with lower odds of occurrence. This approach maximizes the average amount of information contained in each binary digit, effectively compressing the speech signal. By utilizing these compression methods, speech compression can preserve crucial speech qualities while making optimal use of transmission and storage resources. This enables efficient speech signal storage and transmission, reducing memory space requirements, lowering transmission bandwidth, and ensuring the preservation of important speech characteristics.

II. Literature Survey

Various methods and approaches has used of various wavelet types (e.g., Daubechies, Coiflets), different transform techniques (e.g., DCT, DWT), incorporation of coding schemes (e.g., Huffman encoding, quantization), and exploration of additional techniques (e.g., packet wavelets, polynomial approximation, and median function thresholding) are examples of differences in methods used across the papers.

In order to eliminate redundancy in speech signals, the evaluated studies emphasise the significance of effective speech compression techniques. Wavelet Transform is useful for increasing compression ratios while preserving perceptual features, especially when used with Huffman coding. The advantage of Wavelet Transform over DCT in voice compression is highlighted by the exploration of several wavelet types and transform methodologies. The effectiveness and benefits of additional techniques including the Hamming Correction Code Compressor, linear predictive coding, and sub band coding are demonstrated.

III. Objective

1. Develop a speech compression technique that efficiently removes redundant information from speech signals.
2. Investigate the application of Wavelet Transform to decompose the speech signals into high pass and lowpass sub bands, thereby eliminating noise and retaining essential information.
3. Implement Huffman coding to further compress the speech signals, optimizing the average information per binary digit and achieving a higher compression ratio.
4. Evaluate the performance of the proposed technique by assessing the distortion in the reconstructed waveform, measuring the percentage of preserved signal energy, and analysing the perceptual qualities of the compressed speech.

IV. Block Diagram

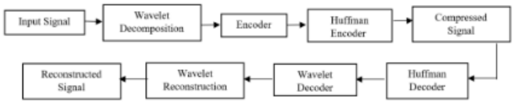


Figure 1. Block Diagram of the Flow of Implementation

V. Methodology

The method of voice compression and reconstruction utilising wavelets and Huffman coding is shown in a flowchart in Figure 1. The Python interpreter in the VS Code IDE was used to implement the entire article. The Wavelet transform is initially used to decompose the input signals. The calculation of a threshold value results in the disregard of DWT coefficients below the threshold. As a result, compression is achieved by concentrating the information into fewer DWT coefficients. The compression works well despite the slight distortion present. The Huffman source coding technique is used by the encoder to compress the signal after encoding the coefficients. The Huffman decoder and then the wavelet decoder are used to decode the compressed signal during the decoding stage. The signal that was rebuilt is a WAV file.

Compression with Wavelets

The db6 wavelet was chosen for this experiment because it had the most energy after the first level of decimation and produced a sizable proportion of zero counts after the fifth level. Using the db6 wavelet, the recorded signals were wavelet decomposed up to the sixth level. There were two parts to each degree of decomposition: an approximate component and a detailed part. The Approximation component was further divided into an Approximation part and a Detailed part, with the Detailed part remaining unaltered. The majority of coefficient values became negligible and could be deleted as a result of this decomposition procedure, which was carried out up to the sixth level. The Python interpreter in the VS Code IDE was used to determine a threshold that would compress the signal. The coefficient values that above the threshold were kept after the thresholding procedure and formed a row matrix to represent the compressed signal. The non-zero elements' indexes, which were crucial for signal processing, were also saved by the encoder.

Compression with Huffman Coding

The research investigates the effectiveness of wavelet transform alone versus wavelet transform combined with Huffman coding for speech compression. The wavelet encoder rounds up non-zero coefficients to the nearest integer, which are then used as symbols in the subsequent Huffman encoding process. The implementation of the Huffman dictionary and encoder is carried out using MATLAB, resulting in binary string outputs. The length of these output signals represents the number of bits required for transmission. By calculating the compression ratio, which compares the original signal size to the compressed signal size, the efficiency of the compression techniques can be assessed. Moreover, distortion analysis is conducted to evaluate the accuracy of the reconstructed waveform in comparison to the original signal.

Compression Ratio=(Length of orignal signal in bits)/(Length of compressed signal in bits)

Reconstruction of original signal

The Python interpreter in the VS Code IDE is used to implement the Huffman and wavelet decoders. The waveform reconstruction is stored as a WAV file so that it can be played back by various audio software programmes. The Python matplotlib module is used to plot the original input signal as well as the reconstructed output signal. Metrics like distortion, quantified as the Mean Square Error (MSE), are calculated to judge the integrity of the reconstructed signal. To gauge the degree of information preservation, the retained signal energy of the rebuilt signal is assessed. The research ensures effective application and evaluation of the compression and reconstruction strategies by utilising the Python interpreter in VS Code.

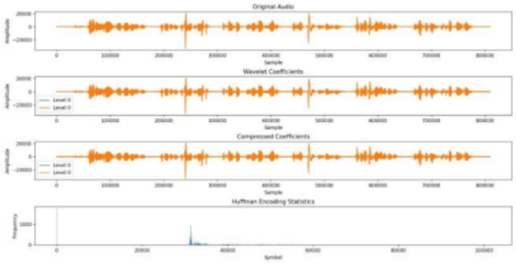
$$Retained\ Energy = \frac{Energy\ in\ reconstructed\ signal}{Energy\ in\ orignal\ signal}$$

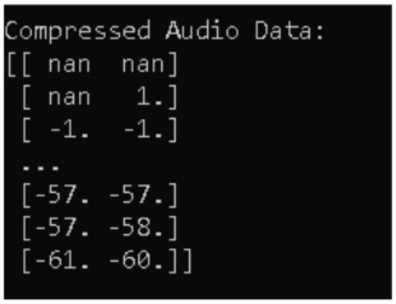
$$Mean\ Square\ Error = \frac{1}{N}\sum_{k=0}^{N}(Rsignal(k) - Osingnal(k))^2$$

N = Total number of samples
Rsignal = Reconstructed Signal
Osignal = Original Signal

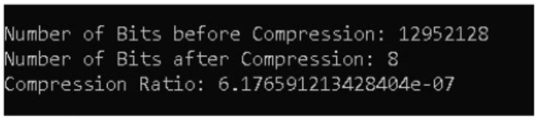
VI. Results

The combination of wavelet transform and Huffman coding achieved effective speech compression with a significant reduction in the number of bits required for transmission. The compression ratio indicated improved efficiency. Distortion analysis showed minor signal degradation, but the reconstructed waveform largely preserved the original signal's integrity.





```
Compressed Audio Data:
[[ nan  nan]
 [ nan   1.]
 [ -1.  -1.]
 ...
 [-57. -57.]
 [-57. -58.]
 [-61. -60.]]
```



```
Number of Bits before Compression: 12952128
Number of Bits after Compression: 8
Compression Ratio: 6.176591213428404e-07
```

VII. Conclusion and Future Work

The study demonstrated that using wavelet transform in conjunction with Huffman coding provides an efficient approach for speech compression. Despite slight distortion, the method effectively reduces the signal's size while maintaining important information. The integration of Python interpreter in the VS Code IDE allowed for successful implementation and evaluation of the compression and reconstruction strategies

.References

[1] Speech Compression with Wavelet Transform and Huffman Coding Ranjushree Pal, Department of Electronics and Telecommunication Engineering, Dwarkadas J Sanghvi College of Engineering, Mumbai: 400056, India.

[2] L. L. Motta, R. S. Mendes and M. H. M. Costa, "An Audio Compression Method Based on Wavelet Packet Decomposition, Ordering, and Polynomial Approximation of Expressive Coefficients," 2019 IEEE Latin-American Conference on Communications (LATINCOM), Salvador, Brazil, 2019, pp. 1-6, doi: 10.1109/LATINCOM48065.2019.8937917.

[3] Speech Compression Exploiting Hamming Correction Code Compressor Islam Amro Faculty of Information Technology Al-Quds Open University Ramallah.

[4] A. Debnath, U. K. Mondal, B. Bikash Roy and N. Panja, "Achieving Lossless Audio Encoder through integrated approaches of Wavelet Transform, Quantization and Huffman encoding (LAEIWQH)," 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), Gunupur, India, 2020, pp. 1-5, doi: 10.1109/ICCSEA49143.2020.9132865.

[5] M. S. Serina and S. G. Mosin, "Digital Audio Information Compression Using Wavelets," Proceedings of the International Conference Mixed Design of Integrated Circuits and System, 2006. pp. 660-664, doi: 10.1109/MIXDES.2006.1706666.

[6] M. H. Martínez, N. F. Rosas and C. Camargo, "Compressing audio signals with the use of the wavelet transform and Embedded Systems," Symposium of Signals, Images and Artificial, Bogota, Colombia, 2013, pp. 1-5, doi: 10.1109/STSIVA.2013.6644947.

[7] Wavelet transform for speech compression and denoising Fatma zohra Chelali* and Noureddine. Cherabit, University of Science and Technology Houari Boumediene of Algiers, ALGERIA Amar. Djeradi and Leila.Falek

[8] G. Kemper and Y. Iano, "An Audio Compression Method Based on Wavelets Subband Coding,"

[9] A. K. Sinha, S. K. Dutta, B. Dev, Jaydipta, R. Ranjan and R. Kumari, "Wavelet based Speech Coding technique using median function thresholding," 2014 International Conference on Electronics and Communication Systems (ICECS), Coimbatore, India, 2014, pp. 1-4, doi: 10.1109/ECS.2014.6892538.

[10] A Comparative Study between Discrete Wavelet Transform and Linear Predictive Coding D.Ambika, V.Radha

[11] F. z. Chelali, N. Cherabit, A. Djeradi and L. Falek, "Wavelet transform for speech compression and denoising," 2018 6th International Conference on Multimedia Computing and Systems (ICMCS), Rabat, Morocco, 2018, pp. 1-7, doi: 10.1109/ICMCS.2018.8525996.

[12] Continuous Speech Coding Using Coiflets Wavelet. Shijo M Joseph1 Babu Anto

[13] J. Pang, S. Chauhan and J. M. Bhlodia, "Speech Compression FPGA Design by Using Different Discrete Wavelet Transform Schemes," Advances in Electrical and Electronics Engineering - IAENG Special Edition of the World Congress on Engineering and Computer Science 2008, San Francisco, CA, USA, 2008, pp. 21-29, doi: 10.1109/WCECS.2008.11.

Speech compression using Huffman coding and Wavelet Transform

ORIGINALITY REPORT

12%	3%	12%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1 Ranjushree Pal. "Speech Compression with Wavelet Transform and Huffman Coding", 2021 International Conference on Communication information and Computing Technology (ICCICT), 2021
Publication — 7%

2 www.researchgate.net
Internet Source — 1%

3 Prasad Kumar, Parvathy Venu M, K.M. Sandeep, Veena Shivadas Kindalkar, Ananya Kote M, S.M. Dharmaprakash. "Non-polar a-plane oriented ZnO:Al thin films for optoelectronic applications", Physica B: Condensed Matter, 2021
Publication — 1%

4 moam.info
Internet Source — 1%

5 "Proceedings of International Conference on Wireless Communication", Springer Science and Business Media LLC, 2020 — 1%

Publication

6	Fatma zohra Chelali, Noureddine. Cherabit, Amar. Djeradi, Leila. Falek. "Wavelet transform for speech compression and denoising", 2018 6th International Conference on Multimedia Computing and Systems (ICMCS), 2018 Publication	1 %
7	Ji, Xiangyang, Jizheng Xu, Debin Zhao, Feng Wu, Fernando Pereira, Heung-Yeung Shum, and Andrew G. Tescher. "", Visual Communications and Image Processing 2005, 2005. Publication	1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

Speech compression using Huffman coding and Wavelet Transform

PAGE 1

PAGE 2

PAGE 3