Implementation of an Audio Compression Algorithm in Python

1st Ms. Bhagya Bagal
Department of Electronics and communication
KLS Vishwanathrao Deshpande Institute of Technology
Haliyal

2nd Ms. Bhagyalaxmi Angadi
Department of Electronics and communication
KLS Vishwanathrao Deshpande Institute of Technology
Haliyal

3rd Ms. Farha Karigar
Department of Electronics and Communication
KLS Vishwanathrao Deshpande institute of Technology
Haliyal

4th Ms. Hemavati Sabnis
5th Prof. Suraj Kadli
Department of Electronics and Communication
KLS Vishwanathrao Deshpande
Institute of Technology
Haliyal

***

Abstract: In today's Digital world efficiency of audio is important, In this Project we get reduced file size while maintaining the quality preceived by the human ear. we implemented audio compression using Algorithm in Python which decreases file sizes, maintaining preceived sound quality. Our method uses perceptual coding, transform coding, entropy coding and deep learning. this results in practical, efficient, and scalable audio compression.

INTRODUCTION:

In this modern era audio data has become an important part of everyday technology for streaming music, videocall, zoom meetings and many more. For this process audio quality is important, the efficiency of audio quality is improving day by day. The original audio which is shared by others or recorded may contain large amount of space, storage and high bandwidth which increases use of more mobile data.

The quick growth of technology and digital network such as music streaming, mobile communication, videocalls, online meetings, online shopping, has increased the demand for maintaining audio quality. The uncompressed audio may contain more storage, space, high bandwidth, which creates problems in user device. Audio compression reduces unnecessary and removes audio that are not good for human hearing, which leads smaller file size which is acceptable. The current development in audio compression uses advance signal processing, Deep learning and its types like Convolutional Neural Network and Transform Coding.

The DCT is a device which converts time-domain audio signal into frequency-domain signal, DCT gives the lowest frequency signals in which the higher frequency signal is removed without effecting the original audio. The CNN is a type of Deep Learning which directly takes the important data given in the audio. it gives the compressed file size form of audio which is reconstructed without loss of original audio. It mainly focuses on implementing audio compression using python and its Libraries like NumPy, SciPy, Sound file, and Matplotlib for compressing.

LITERATURE SURVEY:

In recent years the audio compression is essential due to improving technologies, in this project we discussed some of the literature survey to develop our project, which gives clear quality, less distortion, storing space and less bandwidth which can be implemented in mobiles, laptops.

1. **Fine Tuning Audio Compression Algorithmic implementation and performance metrics:** March (2025) in this paper we get information about the compression algorithm. the key tools or Methods used. This paper gives the information how an audio can be compressed and improved using Fine- tuning techniques. It starts with the challenges

experienced of storing and transmitting of uncompressed audio, which requires more storage, space, and bandwidth. The paper mainly focuses on keeping the important audio and removes unnecessary audio and reduces file size. It keeps the important part and removes the less important audio content and then reconstruct the signal, later gives the output in the form of compression ratio, size reduction, signal to noise ratio and graph. Fine tuning means it converts audio into small frames, it adjusts various parameters and lower the voice according to the human hearing ratio. Python libraries like Python Librosa, Pydub, MATLAB, ML by this report we come to a conclusion that wav-let compression offers exceptional accuracy with minimal Distortion. This survey concludes that performance based fine tuning can increase both efficiency and listening quality of audio, which makes the algorithm suitable for present and future applications.

2. **Audio Feature Extraction Algorithms Implementation Technologies Analysis:** September (2024) In this review we got the required output with high quality audio and compressed size. Audio extraction feature is very important stage in modern audio system, that enables machines to clarify, classify, and compress audio quickly. Several algorithms like spectral centroid, zero crossing rate, spectrograms help convert original audio with high distortion into meaningful and distortion less audio. These features capture important characteristic like frequency content, energy distribution, and short-term variations. Implementation tools have involved from traditional DSP methods to advanced python-based libraries like Librosa, MEL, spectrum, CHROMA spectrum, NumPy, and SciPy, which simplify audio compression into visualization, by this we get essential for effective compression and enhances performance of audio application. The analysis of these algorithms shows that classical methods are lightweight and suitable for real-time embedded devices, while deep learning-based extraction offers superior accuracy for tasks like speech recognition, music classification, and audio compression. Overall, the integration of algorithmic signal processing with modern computational tools enables robust, scalable, and high-performance audio feature extraction across diverse applications.

3. **DSP Real time implementation of audio Compression using Fast Hartley Transform**: September (2018) Real-time audio compression using the Fast Hartley Transform (FHT) provides an efficient and DSP friendly approach for reducing audio data size while preserving essential signal quality. In this process we use Fast Fourier Transform, which uses only real-valued operations, making it lighter and suitable for digital signal. In this method, we got the incoming audio is divided into small frames, and each frame is transferred into the Hartley domain to separate frequency components from less important one. The audio with less efficiency is removed strongly, leading to compression with minimal audio distortion. The compressed audio is framed and transmitted or stored, while the Inverse Fast Fourier transform reconstruct the audio in original time with low delay. this process is useful for systems storage, memory, power consumption are essential, such as portable devices, communication systems. From this we used Libraries like DSP Processor TMS320C6416, Fast HartleyTransform and modern technologies like perceptual coding, in which system meets real time requirements with low complexity. This process of using FHT based audio compression gives a balanced output with a strong audio quality and audible by all.

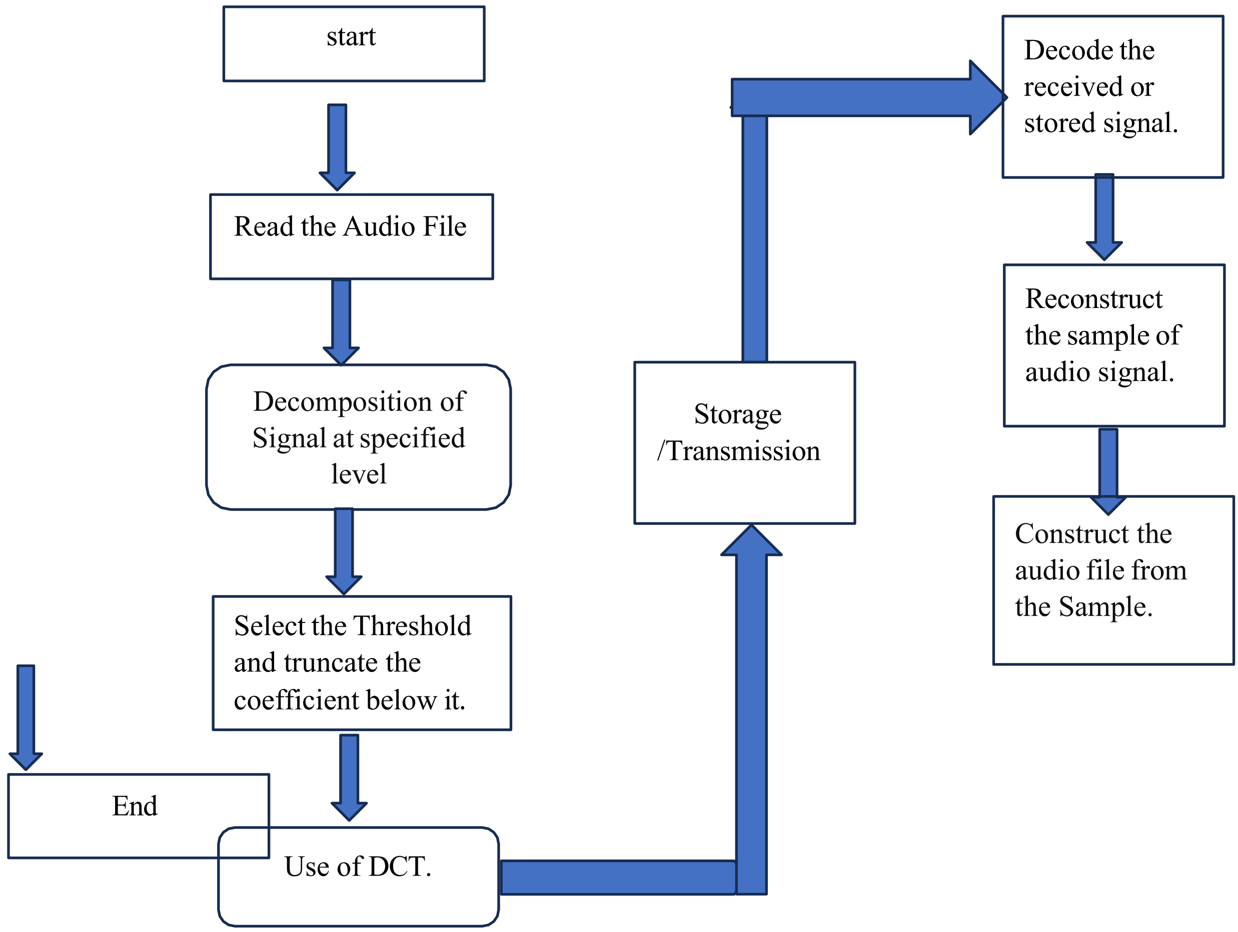
METHODOLOGY:

A. Problem Statement:

The rapid growth in technologies the compressed audio is more important, meanwhile the audio should contain less storage, bandwidth. The uncompressed audio requires large storage area, high bandwidth, transmission requires more mobile data. There-fore transmission of large audio becomes critical, so the compression is required. This project mainly focuses on file size, low bandwidth, and python-based audio compression that provides most suitable compression ratio and audio quality. This approach focuses on identifying higher frequency audio and noise from background. By use of DCT, CNN, Transform deep learning we get the compressed audio in the form of size reduction ratio and graph which shows the variations and difference between original audio and reconstructed audio.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 01 | Jan - 2026 SJIF Rating: 8.586 ISSN: 2582-3930

B. Proposed Method:



In this project the process adopted shows the step by step, rule-based approach that relies the compression of audio without distortion. The main goal is to get compressed audio without distortion, with the use of various python libraries, DCT, CNN, Transform coding. The complete flowchart is discussed below.

The system starts with the start button which starts with giving .wav file later it read the audio file and converts it into digital signal using python and normalize it. Then the signals are made into small frames and quantize the frames. Decomposition of signal at specified level, later select the threshold and shorten the coefficient below the threshold, Later the next step is use of (DCT) Discrete cosine transform which converts audio from time domain to frequency domain. It encodes the audio and further step is storage or transmission in this step the encoded audio or compressed audio is stored and transmitted. The next process follows the reconstruction method which gives the audio in compressed form it decodes and dequantize the received signal during Decompression then the further step is to reconstruct the samples of audio signal next the constructed and compressed audio file from the samples, it gives the compressed audio file which helps to get more storage and more efficiency of audio.

In this process we use several coding like perceptual coding which compress audio to the human hearing as the human ears are sensitive, transform coding converts time domain to frequency domain in this the energy is concentrated in a few coefficients making it easier to identify and remove less important information.

RESULT:

1. Digital Signal Processing based Audio Compression:

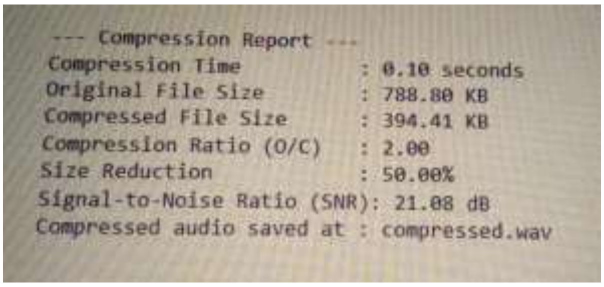


Fig 1

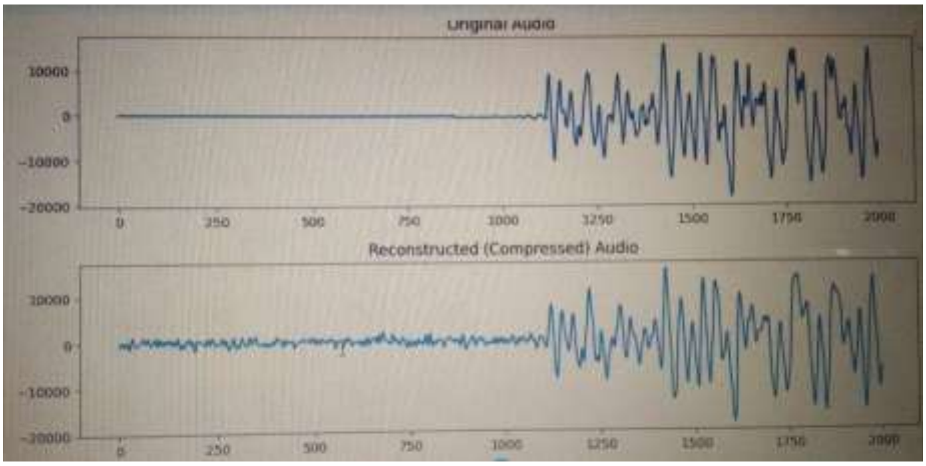


Fig 2

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 01 | Jan - 2026 SJIF Rating: 8.586 ISSN: 2582-3930

2. CNN (Convolutional Neural Network):

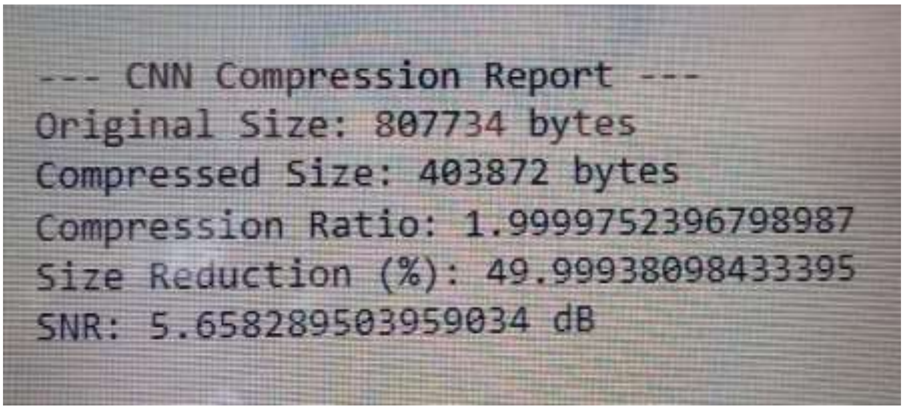

--- CNN Compression Report ---
Original Size: 807734 bytes
Compressed Size: 403872 bytes
Compression Ratio: 1.9999752396798987
Size Reduction (%): 49.99938098433395
SNR: 5.658289503959034 dB

Fig 3

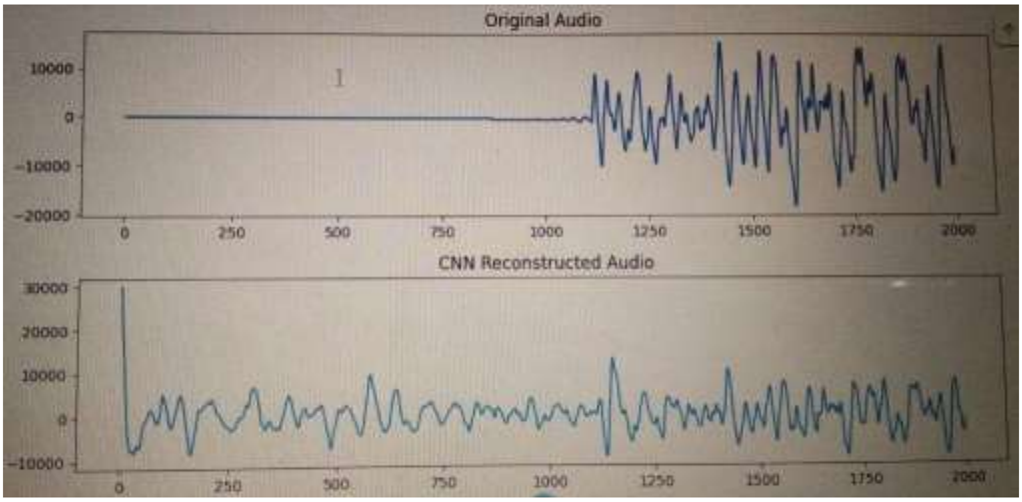


Fig 4

3. Transform Coding:

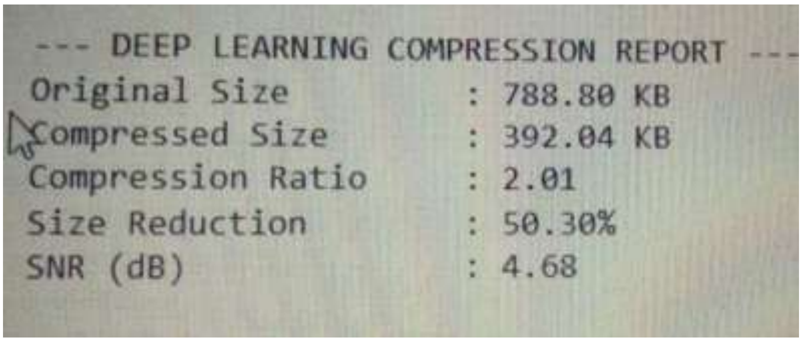

--- DEEP LEARNING COMPRESSION REPORT ---
Original Size : 788.80 KB
Compressed Size : 392.04 KB
Compression Ratio : 2.01
Size Reduction : 50.30%
SNR (dB) : 4.68

Fig 5

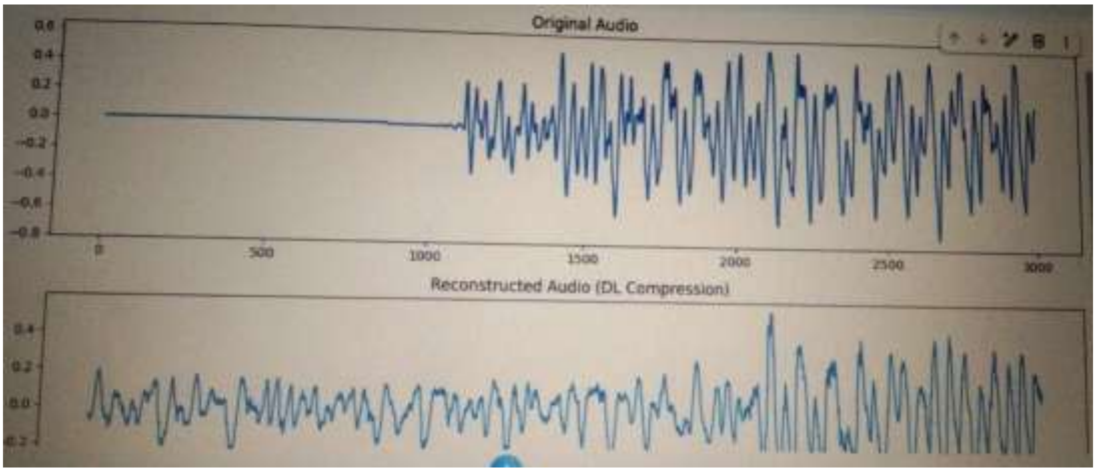


Fig 6

4. CNN:

```
--- CNN Compression Report ---
Original Size: 807734 bytes
Compressed Size: 102573 bytes
Compression Ratio: 7.874723367747848
Size Reduction (%): 87.30114121728192
SNR: 5.96151954135919 dB
```

Fig 7

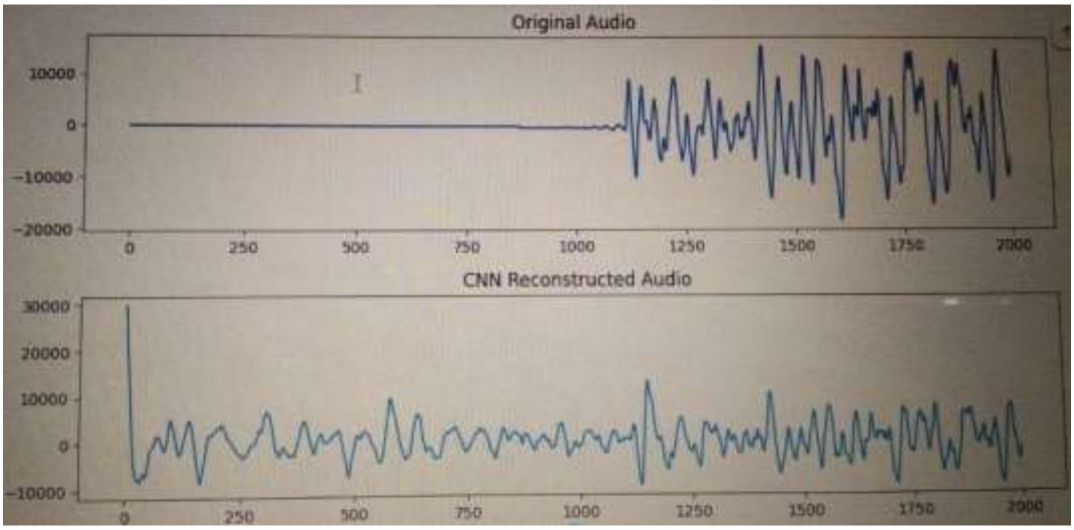


Fig 8

CONCLUSION:

The work carried out in the project describes how the audio compression can greatly improve the handling, storage, and transmission of sound data. By applying Python-based signal-processing techniques, Deep learning and its types the algorithm was able to significantly reduce the file size while keeping the audio quality at an acceptable level for listeners. The results show that even a lightweight compression model can deliver stable performance without adding unwanted noise or distortion. This makes the approach suitable for real-time applications, mobile platforms, and environments with limited bandwidth. Moreover, the implementation process helped in understanding how transform-based methods reduces complex audio signals into manageable components. The successful execution of compression and reconstruction proves the reliability of the chosen method, Finally the project describes that practical and efficient audio compression can be achieved using simple algorithms, making it a strong base and useful for future development such as adaptive compression, machine- learning-based optimization, and integration into real-time systems.

REFERENCES:

[1]. Ijaz. U, Gillani. F, Iqbal. A, Sharif. M. S, Anwar. M. F, Ijaz. A, “Fine- Tuning Audio Compression: Algorithmic Implementation and Performance Metrics”, International Journal of Innovative Science and Technology, Vol.6 Issue. 1pp 220-236, March 2024.

[2]. Shaoxuan. Li, “Audio Feature and Extraction Algorithms and Implementation Technologies Analysis”, pp, 507-516, May 31, 2024, doi:10.1109/ISPDS62779.2024.10667490.

[3]. Souha BOUSSELMI, Noureddine ALOUI and Adnen CHERIF “DSP Real Time Implementation of an Audio Compression Algorithm by using the Fast Hartley Transform”, International Journal of Advanced Computer Science and Application, Vol. 8, No 4, 2017.