Chapter - 3

Audio Watermarking Using Basic Neural Network and Deep Learning Techniques

CHAPTER - 3

AUDIO WATERMARKING USING BASIC NEURAL NETWORK AND DEEP LEARNING TECHNIQUES

3.1 Chapter Introduction:

This research introduces an advanced digital audio watermarking system to enhance the security of digital data, particularly in the context of ownership and copyright protection. Traditional extracting processes in audio watermarking often face limitations and low reliability against various attacks. To address these challenges, a deep learning-based approach is proposed, integrating the DWT and an optimized deep Convolutional neural network (DCNN). The notable contribution lies in the DCNN role in selecting optimal embedding locations, a task crucial for robust watermarking. Hyper parameter tuning, achieved through search location optimization, minimizes errors in the classifier. Experimental results demonstrate superior performance, with the proposed model achieving a Bit Error Rate (BER) of 0.082, Mean Square Error (MSE) of 0.099, and Signal-to-noise ratio (SNR) of 45.363. This outperforms existing watermarking models and showcases the effectiveness of neural network architectures, particularly the DCNN, in optimizing watermark embedding and extraction with minimal bit error. The research also discusses various hybrid and novel techniques in digital audio watermarking, emphasizing the utility of neural networks in advancing the field.

3.1.1 Research motivation:

The motivation behind this research stems from the critical need for advanced and effective digital audio watermarking techniques that prioritize imperceptibility, security, and robustness in an era where digital content faces increasing threats of unauthorized use and piracy. As the ubiquity of digital audio continues to grow, ensuring the integrity and protection of intellectual property becomes paramount. The proposed hybrid approach, combining human locating characteristics and creature searching behavior in the optimization process, aims to enhance the precision of watermark embedding, addressing a gap in existing methodologies. By leveraging DWT for preprocessing and extraction, the research seeks to contribute to the development of sophisticated and resilient digital audio watermarking systems, fostering trust in the secure transmission and utilization of audio content across various platforms and applications. Ultimately, this work strives to offer innovative solutions to the challenges posed by unauthorized access and distribution of digital audio, thereby safeguarding the intellectual property rights of content creators and providers.

i) Audio watermarking:

Audio watermarking is essential for several reasons in the digital landscape. Firstly, it serves as a robust mechanism for copyright protection, allowing content creators and owners to assert ownership of their intellectual property. By embedding imperceptible watermarks into audio signals, creators can trace and prove ownership, deterring unauthorized use and protecting against piracy. Additionally, audio watermarking enhances content authentication and ensures the integrity of audio files. In the era of digital distribution and online streaming, where content can be easily replicated and disseminated, watermarking becomes a crucial tool for maintaining the trustworthiness and authenticity of audio content. This technology also facilitates ownership verification, supporting legal claims in cases of copyright infringement. Overall, audio watermarking is indispensable for safeguarding intellectual property, maintaining content integrity, and reinforcing the security of digital audio in diverse applications and industries. These watermarks are typically encoded as subtle modifications in the audio data, ensuring that they remain perceptually indistinguishable to human listeners. The process involves dividing the audio signal into segments, applying mathematical transformations like DWT for efficient manipulation, and utilizing advanced algorithms, often based on neural networks, to embed and extract the watermark. The embedded information serves as a digital signature, enabling the tracking and authentication of the audio content throughout its lifecycle. Audio watermarking plays a crucial role in addressing intellectual property concerns, enabling content owners to assert ownership, trace unauthorized use, and maintain the integrity of digital audio in a variety of applications, from broadcast media to online streaming platforms.

ii) Purpose of audio water marking: Audio watermarking serves a crucial role in the realm of digital audio content by embedding imperceptible information within audio signals. The primary purpose is to address issues related to copyright protection, content authentication, and ownership verification. By incorporating unique identifiers or codes into the audio data, audio watermarking enables content creators to assert ownership and trace the origin of their work. This technology is particularly valuable for combating piracy, as it deters unauthorized copying and distribution while providing a means to identify and take action against infringing activities. Additionally, audio watermarking finds applications in broadcast monitoring, forensic analysis, and metadata embedding, contributing to a range of functionalities such as tracking the usage of audio content, ensuring its authenticity, and enriching associated metadata for improved management. Overall, audio watermarking plays a pivotal role in safeguarding intellectual property, enhancing content integrity, and supporting various aspects of audio content management and distribution.

3.2 Basic Neural Network Model for audio watermarking:

Machine learning and neural network based methods play a crucial role in audio watermarking, contributing to the development of robust and adaptive techniques for embedding and extracting watermarks in digital audio signals. The process begins with the extraction of relevant features from both the host audio signal and the watermark, forming the basis for the machine learning model. This model, often implemented as a neural network, undergoes training using a dataset containing paired audio signals and corresponding watermarks. During the embedding phase, the trained model seamlessly integrates the watermark into the audio while preserving its quality. In the extraction phase, the same model is employed to accurately detect and retrieve the embedded watermark from the watermarked audio. Machine learning enables the watermarking system to adapt to diverse conditions and resist common attacks, ensuring robustness against signal processing operations and compression. The performance of the system is assessed through metrics such as imperceptibility and security. Overall, machine learning enhances the sophistication and efficiency of audio watermarking, offering an intelligent and adaptive solution for protecting intellectual property and maintaining content integrity.

3.2.1 Methodology:

In the audio watermarking framework, the process is conducted in two essential steps: Embedding and Extraction. Here utilized a 90-minute lecture audio file as the host signal and a binary image of size 180x180 as the watermark. Employing a Back propagation Neural Network (BPNN) in conjunction with the DWT, the framework takes two input sources: the binary water mark image and the host audio signal. To enhance processing speed, the 90-minute audio file is segmented into smaller chunks, allowing flexibility in size ranging from 10 seconds to 5 minutes. The 2D binary watermark image undergoes preprocessing using Open CV, ensuring compatibility with the system. For the host audio signal, apply DWT as a standard and efficient preprocessing technique to extract lower frequency components. The DWT is individually applied to these chunks, and frame-wise analysis is performed. Our custom-built back propagation neural network (BPNN) then embeds the binary bits of the watermark into the audio signal. During decoding, a second neural network extracts the watermark, and DWT is reapplied to the 10-second chunks. These chunks are subsequently converged, reconstructing the original audio file.

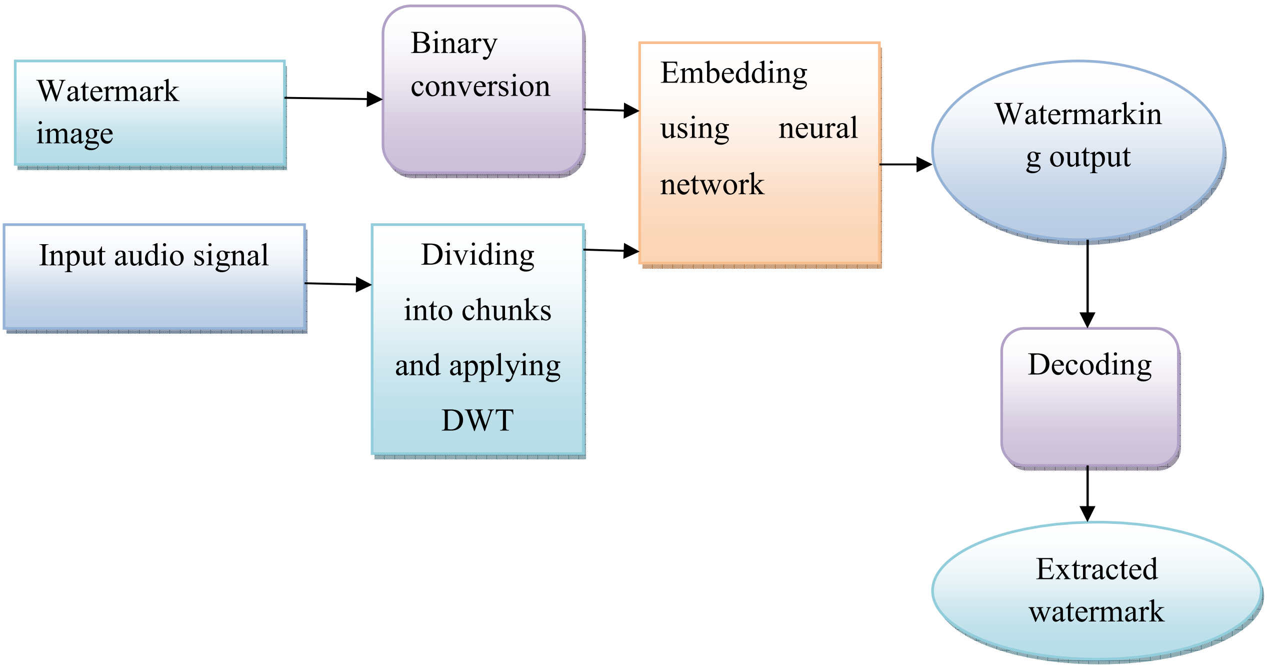


Figure 3.1: illustrative representation of the working of NN in watermarking

i) Input: The first step in the system is to process the watermark image. For the proposed approach here taken the JPG image as sample watermark with the dimensions of 180 x 180 pixels. The JPG image is converted into binary format which is embedded in the audio signal. First convert it into Grayscale by setting a threshold of 128. Further, we had converted the image into binary of size 50 by 50 to use as the watermark. All these preprocessing steps are carried out using cv2 and numpy. Hence the watermark is hereby converted into a numpy array of 50 x 50 dimensions. The audio file that has to be watermarked is first sampled at rate of 44.1 KHz. We down sample the given audio file to 16 KHz using an external library librosa. We divided the whole input audio file of 90 mins into chunks of length 10 seconds to 5 minutes for testing the experimental results. Hence we have the input watermark in the binary form and the audio down sampled into 16 KHz.

ii) Embedding and DWT methods:

In the context of audio watermarking, embedding methods and the DWT are instrumental techniques used to seamlessly integrate imperceptible information into audio signals.

a) Embedding Methods: Embedding involves the incorporation of a watermark into the host audio signal. One common method is to modify the amplitudes or phase of the audio samples to encode the watermark. Spread spectrum techniques distribute the watermark across the entire frequency spectrum, making it less susceptible to removal or distortion. Frequency domain methods, such as modifying coefficients in the Fourier transform domain, are also employed. The goal is to introduce changes that are imperceptible to the human ear while allowing for reliable extraction during watermark detection.

b) DWT: The DWT is a widely used signal processing technique that decomposes a signal into different frequency components through a series of wavelet transformations. In audio watermarking, DWT is often employed to enhance robustness and security. The audio signal is decomposed into approximation and detail coefficients across multiple scales. Watermark information is then embedded in these coefficients, typically in the lower-frequency and less perceptually sensitive bands. This allows the watermark to be robust against various signal manipulations and attacks. The multi resolution analysis provided by DWT facilitates a more effective balance between imperceptibility and robustness in the embedded watermark. The combination of embedding methods, such as spread spectrum or frequency domain techniques, with the transformative power of DWT enhances the overall performance of audio watermarking systems. It enables the incorporation of imperceptible watermarks while providing resilience against common signal processing operations and attacks, making the watermarking process effective for purposes such as copyright protection and content authentication in the audio domain.

iii) Decoding methods: Decoding methods in the audio watermarking refer to the techniques employed to extract the embedded watermark from a watermarked audio signal. These methods are crucial for verifying ownership, ensuring content authenticity, and detecting any alterations or unauthorized use. Two common decoding methods involve correlation-based detection and extraction through the DWT.

a) Correlation-Based Detection: In correlation-based decoding, the watermarked audio signal undergoes correlation with a reference watermark. The reference watermark is a copy of the original watermark that is known to the decoding system. By comparing the correlation values at different points in the audio signal, the system can identify the presence of the watermark. A high correlation indicates a match and suggests that the watermark is present in that portion of the audio. This method is straightforward and often used in scenarios where the watermark is directly recognizable, but it may be sensitive to noise and signal variations.

b) DWT Extraction: The decoding process can also leverage the DWT, especially when DWT was used for embedding. In this method, the watermarked audio signal is subjected to the reverse DWT, known as the inverse DWT or IDWT. By performing this inverse transformation, the original signal is reconstructed while preserving or enhancing the embedded watermark in the process. The watermark information is then extracted from specific frequency bands or coefficients in the transformed signal. This method is advantageous because it aligns with the multi resolution structure of the DWT, enabling selective extraction from different frequency scales and offering robustness against common signal processing operations. Both correlation-based detection and DWT-based extraction methods contribute to the effectiveness of decoding in audio watermarking. The choice of decoding method often depends on the characteristics of the embedded watermark, the robustness requirements, and the specific applications for which the audio watermarking system is designed.

iv) Justification for various neural network and machine learning models in audio watermarking:

The incorporation of machine learning and neural network models in audio watermarking is justified by the need for robust and effective techniques to embed and extract imperceptible information in audio signals, especially for applications such as copyright protection and content authentication. Machine learning models can adaptively learn complex patterns, enhancing the security and resilience of the watermarking process against various attacks and signal processing operations. Various machine learning models find application in different stages of audio watermarking: Machine learning models can be employed during the embedding process to determine optimal locations for watermark insertion, ensuring minimal perceptual impact. Reinforcement learning algorithms, for instance, can adaptively adjust embedding parameters based on perceptual models and signal characteristics, optimizing imperceptibility. During the decoding or watermark extraction phase, machine learning models are crucial for accurately and robustly identifying the embedded watermark. Neural networks, such as deep neural networks (DNNs) or Convolutional neural networks (CNNs), can be trained to recognize watermark patterns, offering improved reliability and resistance to attacks. The adaptive nature of machine learning models allows them to evolve with emerging threats or changes in the audio signal characteristics. Online learning algorithms can continuously update models based on new data, ensuring the watermarking system remains effective over time. Hybrid models, combining traditional signal processing techniques with machine learning, offer a comprehensive solution. For instance, combining the DWT with machine learning models can provide both frequency domain representation and adaptive learning for enhanced performance. In summary, the justification for incorporating various machine learning, deep learning models in audio watermarking lies in their ability to adaptively learn from data, enhance robustness, and improve overall system performance. The diverse array of machine learning models, including reinforcement learning, neural networks, and hybrid approaches, contributes to the effectiveness of audio watermarking for purposes such as copyright protection, content authentication, and ensuring the integrity of audio content.

3.2.3 Neural network – A basic learning model for watermarking:

For the successful embedding of the watermark within the audio file, the Neural Network employs a strategic approach, utilizing the least significant bits (LSBs) of the audio data. Essentially, during the embedding process, the Neural Network adeptly substitutes the LSB of each byte in the audio file with the corresponding bits from the binary watermark data. This substitution is achieved through a combination of logical AND and logical OR operations, providing an efficient means to conceal the secret data, i.e., the watermark, seamlessly within the audio file. General training in neural networks serves the fundamental purpose of enabling the model to learn and adapt to patterns within data. Through a multi-step process, the network adjusts its internal parameters based on the input-output relationships present in a given dataset. This involves forward propagation, where input data passes through the network to produce predictions, and backward propagation, where the model's parameters are updated by minimizing the disparity between its predictions and actual outcomes. The iterative nature of training refines the network's ability to generalize its learned patterns, allowing it to make accurate predictions on new, unseen data. The ultimate objective is to equip the neural network with the capacity to understand the underlying complexities of diverse datasets, fostering its versatility and effectiveness in real-world applications such as image recognition, natural language processing, and other machine learning tasks. In the watermarking process, the neural network performs intricate logical calculations to achieve embedding. With two input features and one output feature, the neural network is designed with a learning rate of 0.01. Employing the Sigmoid function as the activation function, the model undergoes training for 100,000 epochs. This meticulous training regimen ensures the neural network's capacity to adeptly embed the watermark into the audio file, leveraging its learned patterns and logical computations for effective and robust watermarking. Following the encoding process, the audio is decoded to extract the watermark. Initially, the encoded audio is transformed into a byte array. Subsequently, the LSBs are extracted from this byte array, forming a matrix of pixel values representing our watermark image. By visualizing this matrix through plotting, the watermark image can be effectively reconstructed and extracted from the audio, completing the decoding process.

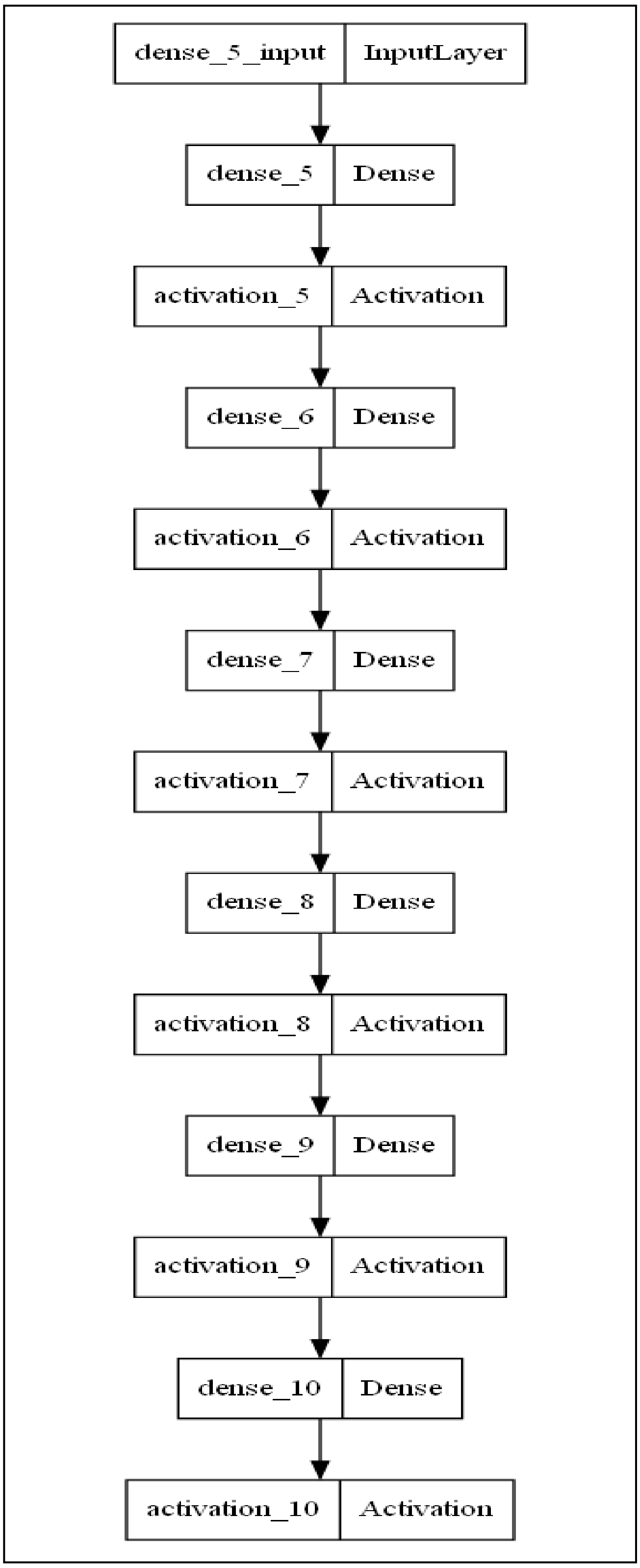


Figure 3.2: Layer details for the NN model

3.3 DCNN with SLOA: Deep learning model for Watermarking.

Deep learning methods play a crucial role in watermarking by offering advanced techniques for embedding and detecting digital watermarks in multimedia content. These methods leverage the CNN model, to enhance the robustness and security of watermarking processes. Deep learning enables the creation of intricate and imperceptible watermarks that can withstand various attacks, ensuring the integrity and ownership of digital content. Additionally, deep learning models can efficiently extract and verify these watermarks, even in the presence of distortions or attempts at removal. The adaptability and learning capabilities of deep learning algorithms make them valuable tools in the development of sophisticated and resilient watermarking solutions, contributing to the protection of intellectual property in the digital domain. In digital audio watermarking, deep learning techniques can be employed for various purposes, including embedding, detecting, and extracting watermarks from audio signals. Deep learning models, offer advantages in terms of robustness, security, and adaptability. Here's an example of how deep learning might be used in digital audio watermarking:

i) Embedding Watermarks: Deep learning models can learn to embed imperceptible watermarks within audio signals. A neural network, possibly a type of auto encoder or a model with specific attention mechanisms, can be trained on pairs of original and watermarked audio samples. The network learns to embed the watermark in a way that minimally affects the perceptual quality of the audio.

ii) Detecting Watermarks: Another application involves using deep learning for detecting the presence of watermarks in audio. A neural network can be trained to distinguish between watermarked and non-watermarked audio signals. This trained model becomes capable of identifying the embedded watermark, even in the presence of various distortions or attempts at removal.

iii) Resistance to Attacks: Deep learning methods contribute to enhancing the robustness of audio watermarking systems. They can learn to withstand common attacks like compression, noise addition, and signal processing manipulations. This adaptability is particularly valuable in real-world scenarios where watermarked audio might undergo unintended alterations.

iv) Content Authentication: Deep learning models can assist in authenticating the integrity of audio content. By learning the unique features of the watermarked signal, these models can verify the presence and correctness of the watermark, ensuring that the audio has not been tampered with. The use of deep learning in digital audio watermarking represents a powerful approach to address challenges related to security, robustness, and anti-tampering in the context of audio content protection.

3.3.1 Methodology: This section provides a comprehensive overview of the pivotal steps within the digital audio watermarking system, with Figure 3.3 illustrating the underlying watermarking models. The research places emphasis on device watermarking techniques aimed at enhancing imperceptibility, security, and robustness. The audio watermarking process unfolds in two distinct phases: the embedding phase and the extraction phase. Initially, the audio signals slated for watermarking undergo segmentation into different signal blocks. The optimal signal within each block is determined through a novel optimization approach called search location optimization, which hybridizes human locating characteristics [41] and the searching behavior [42] of a creature. Subsequently, the smallest block in the audio signal is identified for watermark embedding. Utilizing DWT decomposition, the watermark is seamlessly embedded into the audio signals. During the extraction phase, a 2-level Inverse DWT method is employed to retrieve the hidden message from the watermarked audio file. This methodology ensures a robust and efficient audio watermarking process.

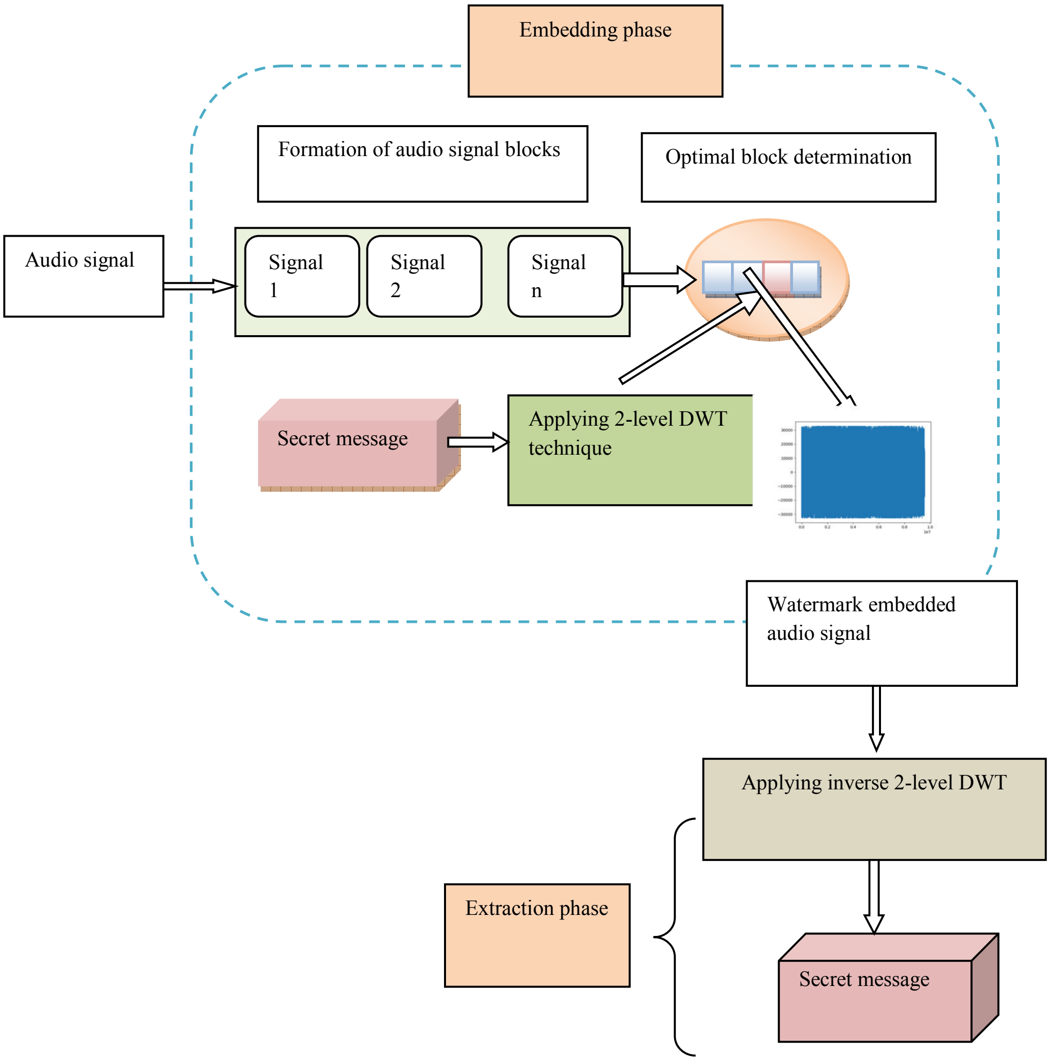

Embedding phase
Formation of audio signal blocks
Optimal block determination
Audio signal
Signal 1
Signal 2
Signal n
Secret message
Applying 2-level DWT technique
Watermark embedded audio signal
Applying inverse 2-level DWT
Extraction phase
Secret message

Figure 3.3: Flow architecture of the proposed digital audio watermarking model

3.3.1.1 Read the input data:

In this research, the secret message for embedding in the audio signal is derived from brain tumour images obtained from the multi-modal brain tumour segmentation dataset of 2020 [43]. Specifically, multimodal MRI scans of lower-grade glioma (LGG) and glioblastoma, subject to pathologically confined diagnoses, are utilized after undergoing pre-processing techniques such as skull-stripping and pixel interpolation. The chosen carrier signal for embedding is an audio signal with a bit rate of 1411 Kbps, lasting for duration of 00:02:02 seconds and stored as a WAV file, resulting in a file size of 20.5 MB. The decision to select an audio signal with a higher bit rate is informed by considerations of quality, bandwidth, and compatibility with existing methods, with the higher bit rate offering superior audio quality and accuracy. The embedding and extraction phases employ a two-level DWT and a two-level Inverse DWT, respectively, ensuring the secure extraction of the secret message without compromising size and quality. The embedding location is determined by a DCNN classifier, leveraging the proposed search location optimization to minimize errors in classification. This approach ensures a robust and accurate process for concealing the brain tumour images within the audio signal.

3.3.1.2 Embedding phase

During the embedding phase, the watermarked digital signal is meticulously generated by incorporating both the host signal and the input watermarks. This process involves the formation of audio signal blocks and the subsequent embedding of secret data, which are thoroughly detailed in the following sections. The host signal, enriched with the concealed watermark, is strategically crafted to ensure the seamless integration of the secret information into the digital signal. These steps contribute to the robust and effective generation of the watermarked digital signal, marking a crucial stage in the overall digital audio watermarking process.

i) Formation of audio signal blocks

In the context of embedding a secret medical image within an audio signal, let's designate the audio signal asT_{audio}, serving as the cover signal. To initiate the embedding process, the carrier audio signal undergoes a random interval partitioning. During this step, the cover signal is segmented into a set of i frames.

$$T_{audio} = T_{audio,t}(i);(1 \le t \le T_{tot})(1 \le i \le U_{tot}) \qquad (3.1)$$

Where, T_{tot} signifies the entire signal duration, while U_{tot}represents the total time interval. The t^{th} signal undergoes a random interval split-up during block formation, denoted as:

$$T_{audio,t}(i) = \{block_1^t, block_2^t, ...block_{a_{tot}}^t\}(t \le u \le a_{tot}) \qquad (3.2)$$

Here, a_{tot} signifies the total number of signal blocks, and the Wavelet transform is employed with two-level decomposition.

3.3.1.3 Secret data embedding using DWT

The DWT is harnessed in the embedding and extraction of audio signals as secret data during the data transaction process. Leveraging its fast computational capabilities and ability to capture intricate details about the data, the DWT proves to be an effective tool. Once the optimal embedding location is determined through the search location optimization process, the DWT is employed to execute the embedding and extraction actions, utilizing the wavelet coefficients to seamlessly conceal and retrieve information from the audio signals. This integration of DWT ensures a robust and efficient process for embedding and extracting secret data within the audio signal.

DWT in the embedding process

At the onset of the embedding process, the original audio signal, sized$a \times b$, is encapsulated by the secret signal, denoted as δec with the size$c \times d$. To initiate the process, the band information of the audio signals is extracted using the wavelet transform. The wavelet transforms are obtained at two levels, distinguishing between High and Low bands. These bands further break down into sub-bands: High-High (HH), High-Low (HL), Low-High (LH), and Low-Low (LL). These sub-bands serve to capture the edge information of the image. The sub-bands at the initial level are designated as

$$Cov_{sig} = \left\{ \mathrm{B}_{\mathrm{HH}}^{sub}, \mathrm{B}_{\mathrm{H}L}^{sub}, \mathrm{B}_{L\mathrm{H}}^{sub}, \mathrm{B}_{LL}^{sub} \right\} \tag{3.3}$$

Here, Cov_{sig} represents the cover signal, and $\mathrm{B}_{\mathrm{HH}}^{sub}, \mathrm{B}_{\mathrm{H}L}^{sub}, \mathrm{B}_{L\mathrm{H}}^{sub}, \mathrm{B}_{LL}^{sub}$ signifies the sub-band in the image with $\mathrm{HH}, \mathrm{H}L, L\mathrm{H}, LL$ coefficients. The dimensions of these sub-bands are denoted as $\left[\frac{c}{2} \times \frac{d}{2}\right]$ following this, these initial sub-bands undergo further processing, progressing to the second level, ultimately generating 16 sub-bands represented by:

$$Cov_{\mathrm{HH}}^{sig} = \left\{ \mathrm{B}_{\mathrm{HH1},} \mathrm{B}_{\mathrm{H}L1}, \mathrm{B}_{L\mathrm{H1}}, \mathrm{B}_{LL1} \right\} \tag{3.4}$$

$$Cov_{\mathrm{H}L}^{sig} = \left\{ \mathrm{B}_{\mathrm{HH2},} \mathrm{B}_{\mathrm{H}L2}, \mathrm{B}_{L\mathrm{H2}}, \mathrm{B}_{LL2} \right\} \tag{3.5}$$

$$Cov_{LH}^{sig} = \left\{ \mathrm{B}_{\mathrm{HH3,}} \mathrm{B}_{\mathrm{H}L3}, \mathrm{B}_{L\mathrm{H3}}, \mathrm{B}_{LL3} \right\} \tag{3.6}$$

The sub-band dimension is determined by $\left[\frac{a}{4} \times \frac{b}{4}\right]$, and utilizing the wavelet coefficients B_{HH} and B_{LL}, the data embedding process is executed. The representation of the embedding process is as follows:

$$\mathrm{E}_{tl}\left(\mathrm{A},\mathrm{X}\right) = \omega_{tl}\left(\mathrm{A},\mathrm{X}\right) + \mathrm{E}_{str} * \mathrm{Sec}_{mbits}^{sig}\left(\mathrm{A},\mathrm{X}\right) \tag{3.7}$$

Where, $\mathrm{E}_{tl}\left(\mathrm{A},\mathrm{X}\right)$ represent the signal embedding process, illustrating the watermarked audio signals. *tl* denotes the total wavelet bands, and *mbits* denotes the total message bits ranging from 1 to 8. The Sec represents the secret message signals, and its wavelet band is denoted as $\omega_{tl}\left(\mathrm{A},\mathrm{X}\right)$, with the variable denoting the embedding strength as E. The embedding process is visually depicted in Figure 3.4

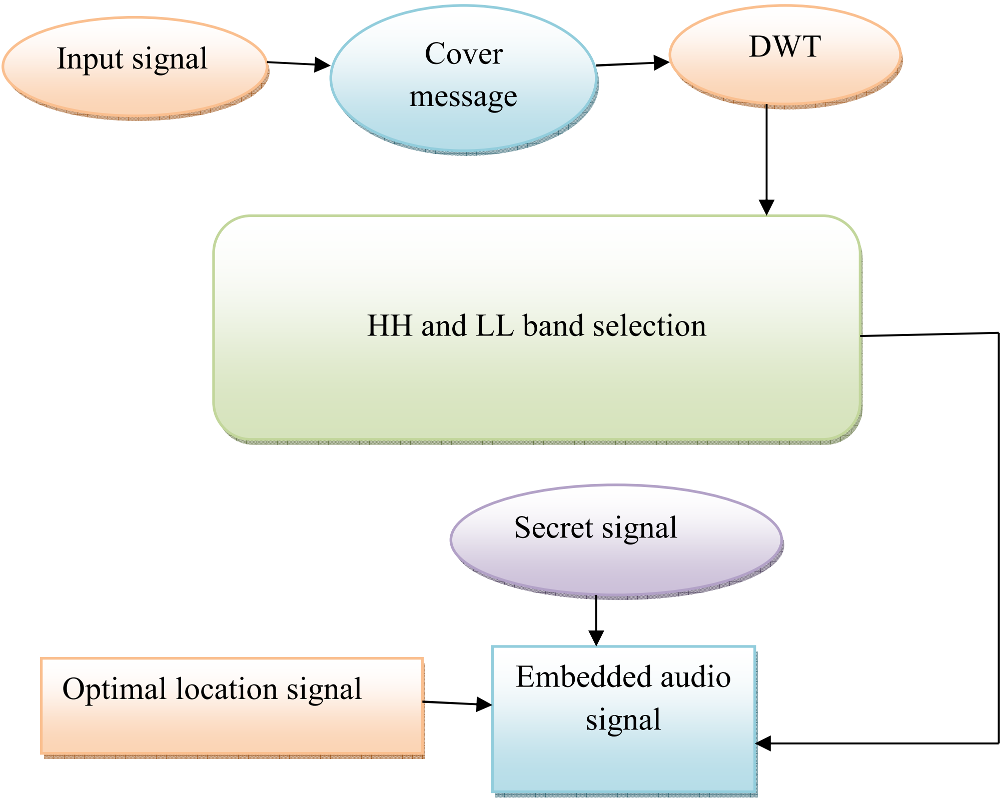


Figure 3.4: The secret data embedding process

3.3.1.4 Extraction phase

During the extraction process, the embedded watermark is retrieved from the watermark audio signal using the inverse wavelet transform. This phase involves a series of steps that will be briefly outlined in this section.

i) Inverse wavelet transforms

The original audio signal is reconstructed from the embedded audio data containing the secret message through the application of the inverse wavelet transform. Analogous to the wavelet transform, the inverse wavelet transform dissects the concealed message into two distinct phases. The decomposition at the initial level is depicted as follows:

$$IDWT\left(\mathrm{E}_{tl}\left(\mathrm{A},\mathrm{X}\right)\right)=\left\{\mathrm{B}^{*}_{LL},\mathrm{B}^{*}_{\mathrm{H}L},\mathrm{B}^{*}_{L\mathrm{H}},\mathrm{B}^{*}_{\mathrm{HH}}\right\} \tag{3.8}$$

The inverse wavelet transforms encapsulate information regarding sub-bands and are denoted as $IDWT\left(\mathrm{E}_{tl}\left(\mathrm{A},\mathrm{X}\right)\right)$. The expression $\mathrm{E}_{tl}\left(\mathrm{A},\mathrm{X}\right)$ signifies the second-level decomposition of the audio signals.

ii) Data extraction: The extraction phase is the inverse process of the embedding phase, aiming to recover the original audio signal from the cover signals. This process relies on key information such as the wavelet-embedded image, optimal point location, and the cover signal. The extraction of secret data is accomplished through the inverse Discrete Wavelet Transform (IDWT). The secret data extraction unfolds in two distinct decomposition levels, as illustrated by:

$$DWT-1\left(\mathrm{E}^{*}\right)=\mathrm{B}_{LL}^{*},\mathrm{B}_{\mathrm{H}L}^{*},\mathrm{B}_{L\mathrm{H}}^{*},\mathrm{B}_{\mathrm{HH}}^{*} \tag{3.9}$$

Where, $DWT-1$ represents the initial level decomposition, while E denotes the representation of the embedded signals.

$$DWT\left(\mathrm{B}_{\mathrm{HH}}^{inv}\right)=\mathrm{B}_{\mathrm{HH}1}^{*},\mathrm{B}_{\mathrm{H}L1}^{*},\mathrm{B}_{L\mathrm{H}1}^{*},\mathrm{B}_{LL1}^{*}$$

(3.10)

$$DWT\left(\mathrm{B}_{\mathrm{H}L}^{inv}\right)=\mathrm{B}_{\mathrm{HH}2}^{*},\mathrm{B}_{\mathrm{H}L2}^{*},\mathrm{B}_{L\mathrm{H}2}^{*},\mathrm{B}_{LL2}^{*} \tag{3.11}$$

$$DWT\left(\mathrm{B}_{L\mathrm{H}}^{inv}\right)=\mathrm{B}_{\mathrm{HH}3}^{*},\mathrm{B}_{\mathrm{H}L3}^{*},\mathrm{B}_{L\mathrm{H}3}^{*},\mathrm{B}_{LL3}^{*} \tag{3.12}$$

$$DWT\left(\mathrm{B}_{LL}^{inv}\right)=\mathrm{B}_{\mathrm{HH}4}^{*},\mathrm{B}_{\mathrm{H}L4}^{*},\mathrm{B}_{L\mathrm{H}4}^{*},\mathrm{B}_{LL4}^{*} \tag{3.13}$$

The optimal location is identified, and the secret data is extracted from the cover audio signal through a mathematical determination process.

$$E_{\sec}\left(\mathrm{A},\mathrm{X}\right)=\omega_{tl}^{*}\left(\mathrm{A},\mathrm{X}\right)-\omega_{LL}^{*}$$

3.4 Existing Deep learning and optimization Models: Discussion and comparison.

In the realm of audio watermarking, the comparison between deep learning, specifically LSTM networks, and existing models highlights the advantages that deep learning architectures bring to this domain. While LSTM models excel in capturing temporal dependencies within sequential audio data, the broader spectrum of deep learning, encompassing CNN and other architectures, introduces additional layers of complexity and adaptability. CNNs, for instance, enhance feature learning by capturing spatial patterns in audio signals. The adaptability of deep learning models proves beneficial in scenarios where diverse audio content and varying conditions necessitate a flexible approach. Furthermore, deep learning's efficient preprocessing capabilities, such as those offered by auto encoders contribute to improved generalization and resilience against common signal processing attacks. In essence, the comparison underscores the potential of deep learning to augment and optimize audio watermarking methodologies, providing a more sophisticated and adaptive approach to the embedding and extraction of watermarks in audio signals.

3.4.1 LSTM:

LSTM, a subtype of Recurrent Neural Networks (RNNs), addresses the limitations of traditional RNNs in effectively handling time sequence data. While RNNs exhibit efficient performance by assigning varying weights to information to internally form a memory of sequential events, they suffer from the vanishing gradient problem during training, impeding the network's ability to retain long-term data memory. LSTM addresses this issue by implementing mechanisms to filter and selectively update the cell state during runtime, ensuring optimal historical memory utilization for accurate predictions. The LSTM model comprises interconnected recursive sub-modules, each encompassing the cell state, output results, and gate structures operating on the state. The gate structures, including the forget gate, input gate, and output gate, play a crucial role in information selection by outputting values between 0 and 1, determining whether to retain or discard information. Specifically, the forget gate erases previous memory, the input gate receives new memory, and the output gate regulates the writing of memory information to other neurons. This dynamic gate mechanism enhances LSTM's ability to capture and retain relevant temporal dependencies. The relevant mathematical formulations governing these processes contribute to LSTM's effectiveness in handling long-term sequential data, offering improved performance in various applications.

$$f_e = \sigma\left(W_f\left[h_{e-1}, y_i\right] + b_f\right) \tag{3.14}$$

$$i_e = \sigma\left(W_i\left[h_{e-1}, y_i\right] + b_i\right) \tag{3.15}$$

$$S_e^{'} = \tanh\left(W\left[h_{e-1}, y_i\right] + b_s\right) \tag{3.16}$$

$$S_e = f_e.S_{e-1} + i_e.S_e^{'} \tag{3.17}$$

$$O_e = \sigma\left(W_o.\left[h_{e-1}, y_e\right] + b_o\right) \tag{3.18}$$

$$h_e = O_e.\tanh\left(S_e\right) \tag{3.19}$$

The LSTM model involves several critical components denoted by different variables and parameters to address the challenges associated with the vanishing gradient problem. Let's delve into the components and expressions within the LSTM architecture. The forget gate result, input gate result, and the output gate consequence, denoted as f_e, i_e, O_e respectively, play crucial roles in determining the flow of information. The cell state at the current moment $S_e, S_e^{'}$ and its updated version are integral to the memory operations of LSTM. Additionally, σ represent different types of neural network activation functions, and the parameters W, b are inherent parts of these activation functions. The overall LSTM output is expressed as h_e. This comprehensive set of formulas effectively captures the information transfer process from the hidden layer at the $e-1$ moment to the hidden layer at the e moment. By introducing gate neurons to manage the cell state, LSTM significantly enhances its ability to perceive information from previous time nodes, effectively mitigating the issue of gradient disappearance during the training process. This advanced architecture contributes to improved information retention and learning capabilities in sequential data processing.

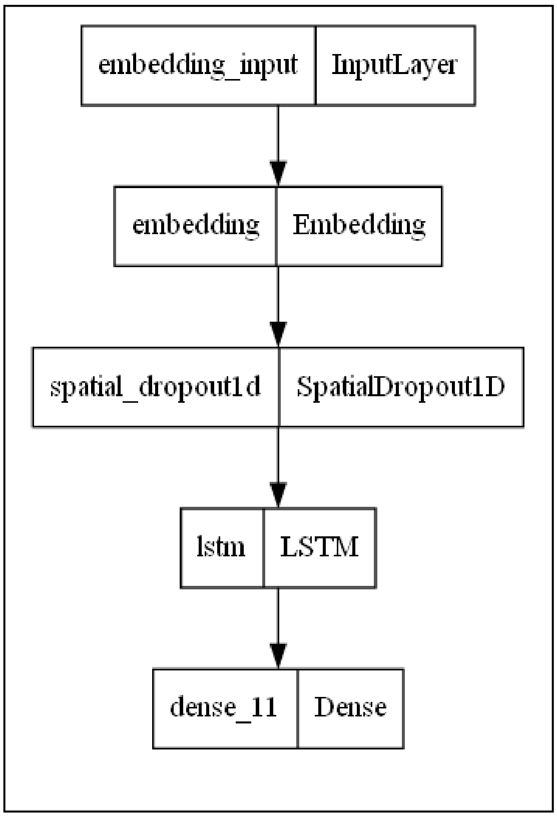


Figure 3.5: Layer details for the LSTM

3.4.2 Deep CNN:

The optimization models effectively tunes the weight and bias of the classifier models, which helps in increasing the accuracy of the models. The optimal blocks for embedding the secret medical message or image are meticulously chosen from the generated signal blocks through the application of search location optimization on a DCNN. DCNN, a widely utilized tool in signal processing, proves effective in selecting the ideal block for secret data embedding. The architecture of the DCNN classifier is visually represented in Figure 3.4, with additional details provided in Table 3.1. This classifier plays a pivotal role in the embedding process, employing advanced Convolutional neural network techniques to identify optimal locations for concealing the secret medical message or image within the audio signal blocks.

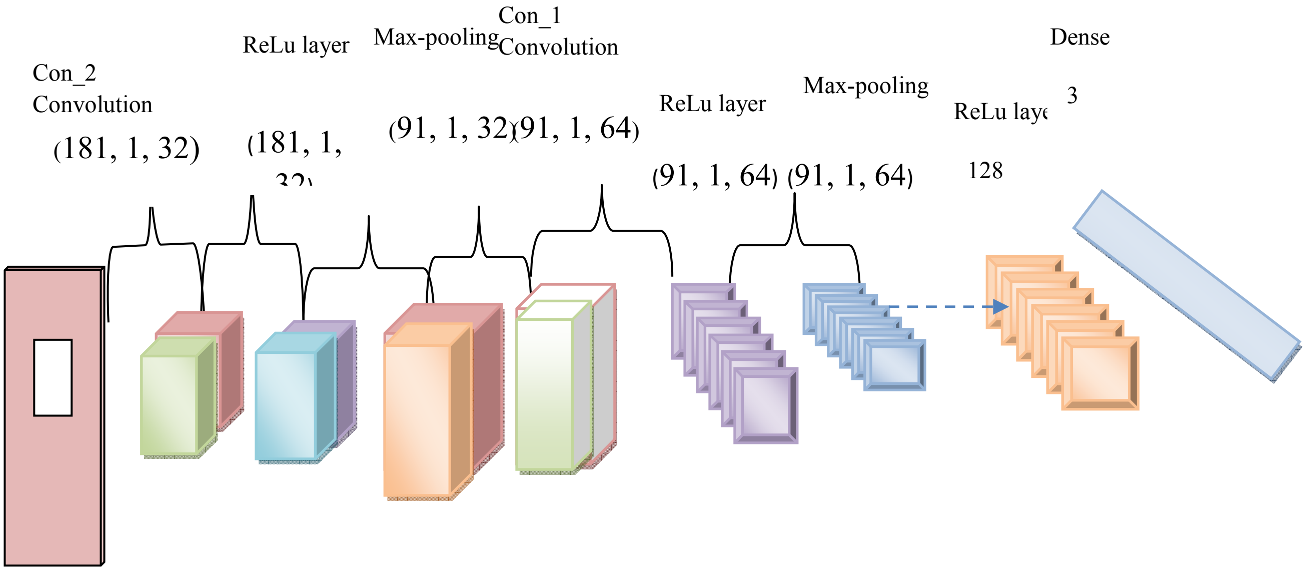


Figure 3.6: Architecture of the deep CNN layer

Table 3.1: Layer information of deep CNN

Layer type	**Output shape**	**Parameters**
conv2d	(None, 181, 1, 32)	544
leaky_re_lu	(None, 181, 1, 32)	0
max_pooling2d	(None, 91, 1, 32)	0
conv2d_1	(None, 91, 1, 64)	18496
leaky_re_lu_1	(None, 91, 1, 64)	0
max_pooling2d_1	(None, 91, 1, 64)	0
conv2d_2	(None, 91, 1, 128)	73856
leaky_re_lu_2	(None, 91, 1, 128)	0

max_pooling2d_2	(None, 91, 1, 128)	0
flatten	(None, 11648)	0
dense	(None, 128)	1491072
leaky_re_lu_3	(None, 128)	0
dense_1	(None, 3)	387

The proposed model is composed of 3 Convolutional layers, 4 leaky ReLu layers, 3 max-pooling layers, 1 flatten layer, and 2 dense layers. The core of the DCNN lies in its Convolutional layers, which generate a feature vector. The first Convolutional layer produces an output feature vector of size (181×1) with a batch size of 32. The subsequent ReLu layer, employing the sigmoid function, scales the CNN and generates an output of(181×1). Following this, the max-pooling layer reduces the spatial size of the input image. This sequence is repeated until a batch size of 128 is achieved, culminating in a single elongated feature vector of size 11648 through the flattening layer. The dense layer categorizes the input into 128 class outputs, scaled by the ReLu layer. Ultimately, the dense layer is employed to obtain 3 class outputs with a batch size of 387, providing a comprehensive architecture for effective signal processing and optimal block selection in the watermark embedding process.

3.4.3 HSO deep CNN: In the context of deep CNNs, the justification likely stems from advancements in computational efficiency or complexity reduction, while still maintaining or improving watermarking performance. The HSO deep CNN model aims to optimize specific aspects of the deep learning architecture, contributing to streamlined processing and potentially enhancing overall effectiveness in audio watermarking tasks.

Hybrid particle swarm optimization: In PSO, two factors cause the swarm to converge prematurely. For the global best of PSO, the fundamental idea is that particles converge to a single point on the line between the pbest and gbest locations, although this point is not guaranteed for a local optimum. Second, the rapid flow of information between particles leads to the generation of comparable particles without increasing population diversity, as well as the ability of the PSO to avoid local optima. In order to get over PSO's drawbacks, it is therefore a good idea to combine the hybrid PSO approach with a local search method. This allows PSO to identify potential solutions where the global optimum may exist while the local search method uses a fine search to locate the global optimum with precision. In addition to preventing the issue of trapping to local optimum by simple local search, this type of solution strategy accelerates the rate of convergence compared to pure global search.

Genetic algorithm (GA) crossover and mutation operators are used in the creation of the HSO algorithm. Due to the fact that the main benefits of the GA above other optimization algorithms are mostly associated with its capacity to avoid local minima and its independence from initialization. Additionally, it is an effective method that can be used to enhance the convergence ratio and choose the algorithm parameters wisely. It performs well across a wide range of study domains. As a result, in every generation, the fitness function values of every individual are determined within the same population, and the PSO scheme initially processes the half of the top performers who exhibit exceptional performance. Here conduct crossover and mutation operators on updated positions of the global best particles of PSO, rather than reproducing the half top ones directly to the next generation. In order to prevent convergence on subpar and wrong solutions and to retain a high degree of population variety of potential solutions, selection operations are typically stochastic in nature. Following that, the probability-based crossover and mutation processes are done to a chosen chromosome. Additionally, the two individuals chosen at random in the tournament selection process are employed. When choosing which parent is superior, their fitness values are compared to the previous higher fitness value. Updated individuals expect the produced offspring to outperform part of the original population, with the weaker performers being eliminated from the population over successive generations. In contrast, PSO is used to inform the highest-ranking members of each generation in order to enhance PSO's capacity to limit and regulate velocities. Since every particle in the swarm gravitates toward better positions, the population as a whole can eventually work together to find the best solution. In the context of deep CNNs, the justification likely stems from advancements in computational efficiency or complexity reduction, while still maintaining or improving watermarking performance. The HSO deep CNN model aims to optimize specific aspects of the deep learning architecture, contributing to streamlined processing and potentially enhancing overall effectiveness in audio watermarking tasks.

3.4.4 Human Mental Search based Optimization Model

It is a population based metaheuristic algorithm which explores the regions around each solution, then grouping for finding the promising region and finally moving towards best strategy. It is mimicking the behavior of the exploration strategies based on online auctions. At the end the auction winner might be having high probability of buying a costly product at the lowest price.

Basic Structure of HMS Algorithm [41].

1. Each participant has a strategy α,
2. Each person can provide a bid,
3. The next bid of every person is consistent with the Levy flight distribution,
4. Multiple bids are allowed,
5. The losing participants try to pick the winner’s strategy for the subsequent auctions.

In this specific type of algorithm each single solution is called a bid, and the Cost value of a bid is obtained by evaluating the cost function [15]:

Cost V alue of a bid = f (bid) = f (x1, x2,...,xNVar)

Here, the mental search represents the number of consecutive values produced for each bid. Few of them are newly created bids around a bid based on Levy flight. Where a Levy flight is a particular type of random walk determining step size with a Levy distribution. Random walk is a Markov chain in which the next position depends only on the current position [41].

3.4.5 Proposed SLOA –deep CNN:

Proposed Search Location Optimization Algorithm (SLOA)

The novel SLOA is inspired by the searching behavior of creatures [42], designed to enhance the searching speed of a creature using locating characteristics [41]. The Search location optimization, which is devised by hybridizing the locating characteristics of humans [41] and the searching behavior of a creature [42]. Drawing on the concept of undetermined optima in optimization problems as open patch-ups dispersed randomly across the exploration area, the followers in the group navigate the area in pursuit of these patches. The algorithm posits that the crucial mutational characteristics of promoters and feeders are similar, allowing them to move interchangeably. At each iteration the follower in the group positioned in the most promising location, providing the best fitness value is selected as the promoter. The promoter then halts, scanning the surroundings for the optimum solution. In the scanning process, the promoter employs eyesight, a primary scanning technique widely preferred by creatures. Different species utilize retinas with adjustable spatial resolutions for visual scanning, enhancing survival by swiftly focusing on potential targets. The SLOA strategy integrates the locating characteristics of humans to maximize fast scanning, minimize iterations, and reduce computational time, thereby improving the algorithm's efficiency.

3.4.5.1 Inspiration

In the SLOA strategy, creatures exhibit distinct roles based on their hunting and foraging behaviors. Promoters and feeders are two primary types[42], with dispersed followers contributing to the exploration dynamics by introducing randomness. The group comprises promoters, feeders, and dispersed followers, each playing a unique role. During a specific period, only one promoter is active for exploration, while the rest are designated as feeders to simplify computation. Notably, the assumption is made that all source feeders may converge towards the source discovered by the promoter. The SLOA strategy draws inspiration from common creature exploration characteristics and employs a communal scanning technique, making it particularly suited for addressing continuous function optimization problems. While the promoter in the SLOA strategy and the globally best particle share similarities, promoters differentiate themselves by retaining the head angle information, a distinctive feature not shared by feeders and dispersed followers. The subsequent section details the mathematical modeling of the SLOA algorithm based on the exploration direction angle, elucidating its application in solving optimization problems for continuous functions.

3.4.5.2 Mathematical modeling of search location optimization algorithm

i) Initialize the positions and head angles

In the j dimensional exploring region, during the v^{th} exploring period, the u^{th} follower has a current location denoted as $K_u^v \in^{\lambda^j}$, and the initial head angles are set as follows:

$$\theta_u^v = \left(\theta_{u_1}^v,......,\theta_{u(j-1)}^v\right) \in \lambda^{j-1} \tag{3.23}$$

The exploration direction of the u^{th} follower is defined by a unit vector, and this direction is quantified by the follower's head angle θ_u^v through a Cartesian coordinate transformation originating from polar form. This transformation is articulated as follows.

$$N_u^v\left(\theta_u^v\right) = \left(n_{u_1}^v,.......,n_{u_j}^v\right) \in \lambda^j \tag{3.24}$$

$$n_{u_1}^v = \prod_{w=1}^{j-1} \cos\left(\theta_{u_w}^v\right) \tag{3.25}$$

$$n_{u_1}^v = \sin\left(\theta_{t(x-1)}^v\right) \cdot \prod_{w=1}^{j-1} \cos\left(\theta_{u_w}^v\right) \tag{3.26}$$

$$n_{u_j}^v = \sin\left(\theta_{u(j-1)}^v\right) \tag{3.27}$$

Consider the v^{th} exploration period, where the achieved exploration direction N, determined by the follower's head angle $\theta_u^v = (\pi/3, \pi/4)$ using equation (3.28), is expressed as:"

$$N_u^v = \left(1/2, \sqrt{6}/4, \sqrt{2}/2\right) \tag{3.28}$$

ii) Promoter scanning

At the outset, the promoter randomly establishes three points within the exploring field for the scanning process. These points encompass the zero-degree, right-side hypercube, and left-side hypercube. The coordinates of the promoter points, denoted as K_q, are expressed through equations (3.29), (3.30), and (3.31):

$$K_y = K_q^v + F_1 w_{\max} N_q^v\left(\theta^v\right) \tag{3.29}$$

$$K_F = K_q^v + F_1 k_{\max} N_q^v\left(\theta^v + F_2 \varphi_{\max} / 2\right) \tag{3.30}$$

$$K_w = K_q^v + F_1 w_{\max} N_q^v\left(\theta^v - F_2 \phi_{\max} / 2\right) \tag{3.31}$$

In this context, the promoter is denoted as K, with K_y representing zero degrees, K_F for the right side, and K_w for the left side. F_1 and F_2 represent random numbers, where F_1 falls within the range of N^1 with a standard deviation of 1 and a mean of 0. The maximum pursuit angle is denoted as max, and F_2 lies within the range of N^{u-1} with $\left(0,1\right)$, representing the maximum search space of the promoter and the target as $\varphi_{\max}$. The movement of the individual towards the global best location is facilitated by adhering to the optimal strategy devised by the promoters. This strategy is expressed as follows:

$$K_q^{v+1} = K_q^v + D^*\left(F * H - K_q^v\right) \tag{3.32}$$

By employing equation (3.32), the promoter scanning phase undergoes modification in its standardized form, incorporating the exceptional locating characteristics of the human. This integration is reflected in equations (3.34) and (3.35), as expressed below:

$$K_y = \frac{1}{2}\left\{K_q^v + F_1 w_{\max} N_q^v\left(\theta^v\right) + K_q^v + D * \left(F * H - K_q^v\right)\right\}$$

(3.34)

$$K_y = \frac{1}{2}\left\{K_q^v(2-D) + F_1 k_{\max} N_q^v\left(\theta^v\right) + D * F * H\right\} \tag{3.35}$$

The global best location is represented as H, and the constant term for the promoter scanning at zero degrees is denoted as D. Similarly, the modification of the promoter scanning phase, contingent on the right side of the hypercube, involves the incorporation of constant parameters $\beta\, and\, \lambda$, expressed as:

$$K_F = \beta\left\{K_q^v + Fk_{\max 1} N_q^v\left(\theta^v + F_2\varphi_{\max}/2\right)\right\} + \lambda\left\{K_q^v + D * \left(F + H - K_q^v\right)\right\} \tag{3.36}$$

$$K_F = \beta K_q^v(\beta + \lambda - D.\lambda) + \beta.F_1 w_{\max} N_q^v\left(\theta^v + F_2\phi_{\max}/2\right) + \lambda.D * F * H \tag{3.37}$$

The enhancement of the left side hypercube in the promoter scanning process is achieved by considering both the global and personal best locations of the human. This formulation is expressed as:

$$K_k = \frac{1}{2}\left\{K_q^v + Fk_{\max} N_l^v\left(\theta^v - F_2\phi_{\max}/2\right) + K_j^v + D * \left(F * H - K_q^v\right) + W * \left(F * V - K_l^v\right)\right\} \tag{3.38}$$

$$K_w = \frac{1}{2}\left\{K_q^v(2 - D - W) + F_1 k_{\max} N_l^v\left(\theta^v - F_2\phi_{\max}/2\right) + (D * F * H) + (W * F * V)\right\} \tag{3.39}$$

In this context, $D\, and\, W$ signify constant terms, where H represents the global best location and v denotes the personal best location of the promoter. This refinement ensures a more accurate identification of the global best location in the promoter scanning phase, incorporating the personal best solution across all angles.

iii) Feeder selection

The promoter identifies an optimal point using a suitable fitness function and subsequently migrates to another point if a better source is identified compared to the previous one. If no better source is found, the promoter remains at the current point by rotating its head to a newly generated angle.

$$\theta^{v+1} = \theta^{v} + F_2\beta_{\max} \qquad (3.40)$$

The maximum rotating angle of the promoter is designated as $\beta_{\max} \in N^1$. Upon the completion of t iterations, in the event that the promoters do not identify the optimal region, they initiate a reset by rotating their heads back to an angle of 0 degrees, as articulated below:

$$\theta^{v+!} = \theta^{v} \qquad (3.41)$$

Where, the term $t \in N^1$ is assumed as constant.

In each iteration a group of followers is designated as feeders, tasked with tracking the promoters to reach sources identified by the promoters themselves. Among the three pivotal phases in the SLOA strategy, the first involves mimicking the region to discern the exploring area of the promoter. This mimicking phase entails stalking other entities without revealing any distinct searching strategy. In the stealing phase, feeders directly acquire sources from the promoters. In the v^{th} iteration, the characteristic of the mimicking region for the u^{th} follower can be conceptualized as a random movement towards the promoter, expressed as follows:

$$K_u^{v+1} = K_u^{v} + F_3 \circ \left(K_l^{v} - K_u^{v}\right) \qquad (3.42)$$

Random walks are acknowledged as a highly effective exploration technique for discovering available sources in a stochastic manner. In the v^{th} iteration, the randomized head angle is generated as θ_u using $\theta^{v+1} = \theta^{v} + F_2\alpha_{\max}$ by selecting a random space and transitioning to a different point, as expressed by:

$$K_u^{v+1} = K_u^{v} + w_u N_u^{v}\left(\theta^{v+1`}\right) \qquad (3.43)$$

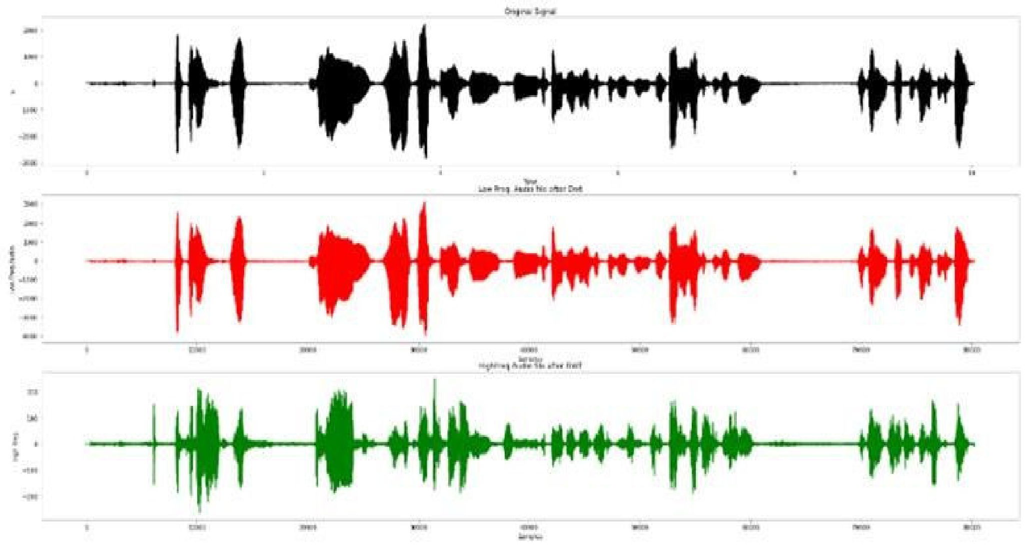
The algorithm SLOA is given as follows,

Algorithm 1. Pseudocode for the proposed SLOA algorithm

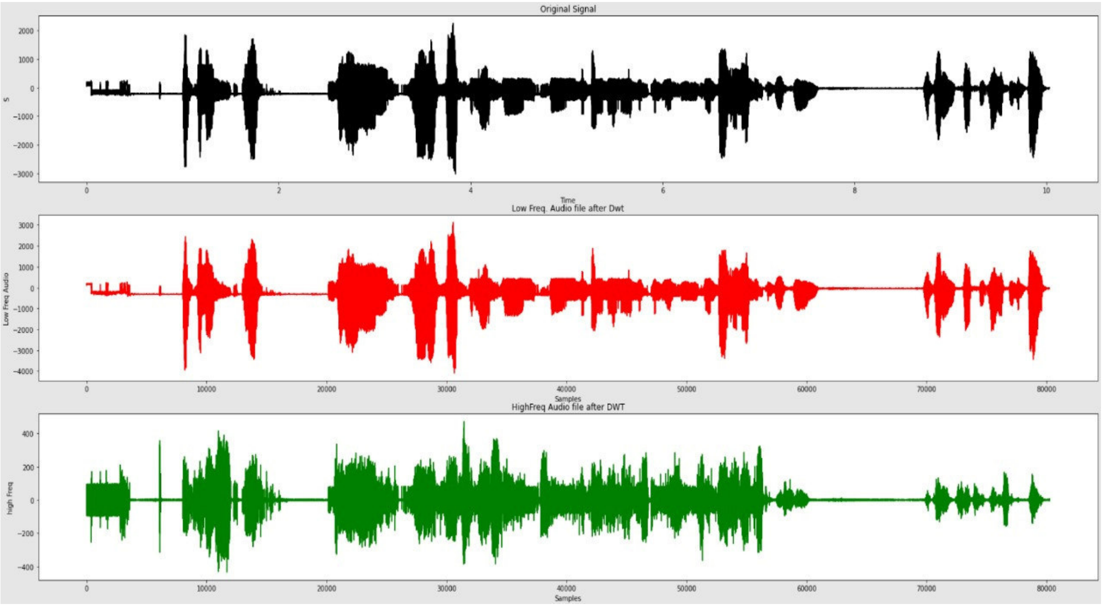
```
Input: $Y_j$
Output: $Y_j^{p+1}$
Initialize positions and head angles
Promoter scanning
    At zero degree based on equation (21)
    On the right side hypercube based on equation (22)
    On the left side hypercube based on equation (23)
Integrate human locating characteristics in equations (21), (22), and (23)
    Obtain the global best location at zero degrees in equation (26)
    Obtain the global best location at the right side hypercube in equation (28)
    Obtain global best location and personal best location at left side hypercube in equation (30)
Feeder selection
    Based on the fitness of $j^{th}$ followers in a group
Random walk as $Y_j^{p+1}$
End while
```

3.5 Results and Discussion:

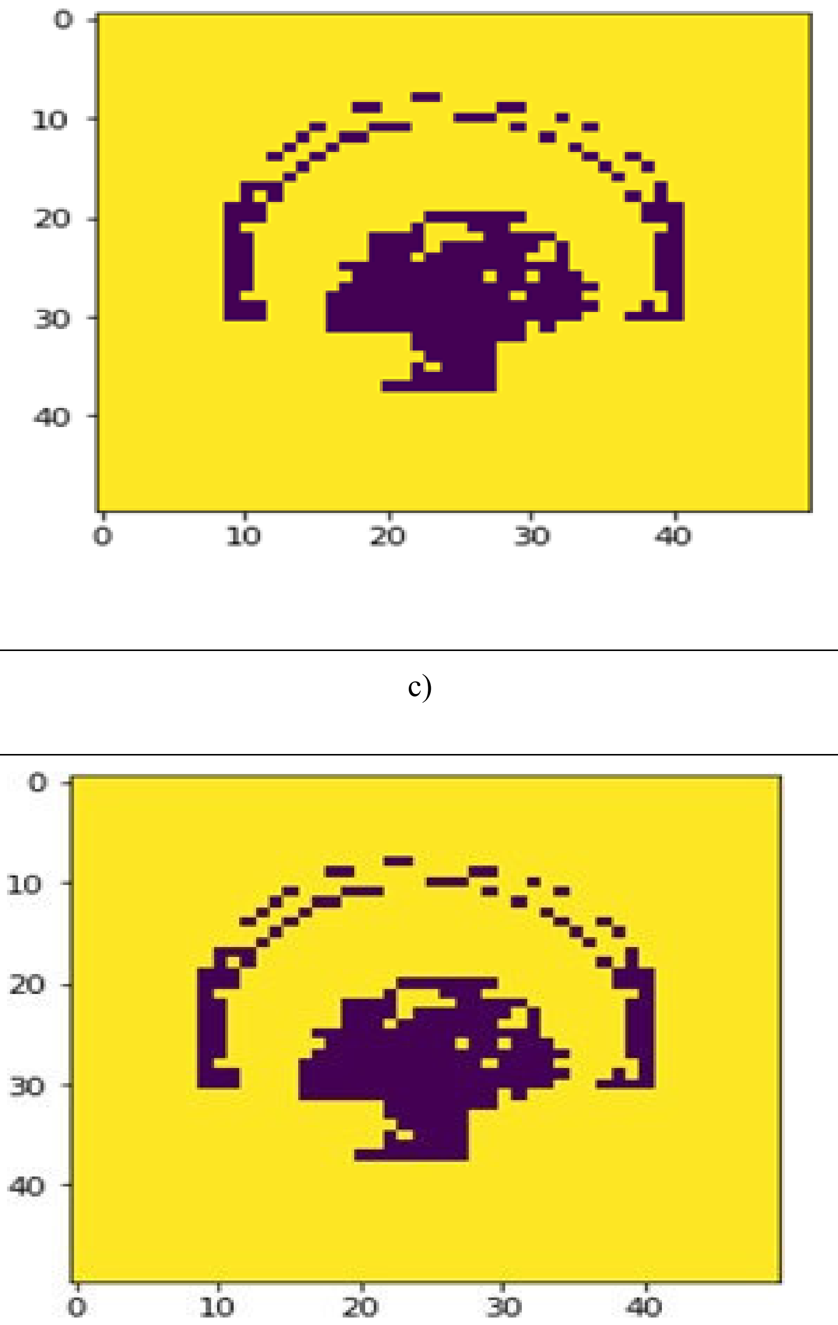
3.5.1 Experimental Result for Basic NN Model: Neural Network (BPNN) in conjunction with the DWT.



a)



b)



c)

d)

Figure 3.7: Experimental result obtained using NN model a) Audio Signal before applying DWT, b) Audio Signal after applying DWT c) Sample Watermark Image and d) Extracted Watermark Image.

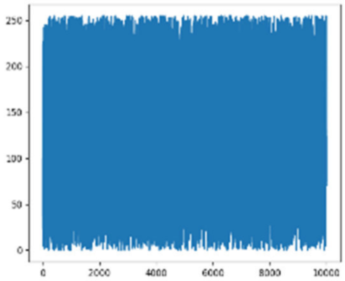
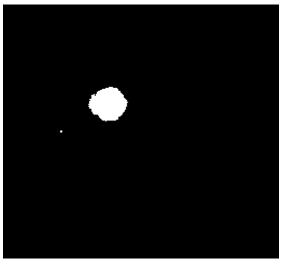
3.5.2 Experimental setup for proposed SLOA.

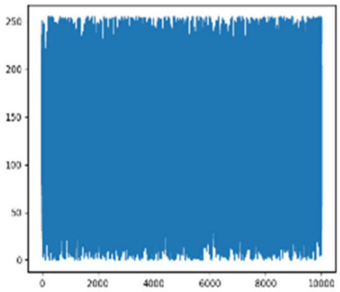
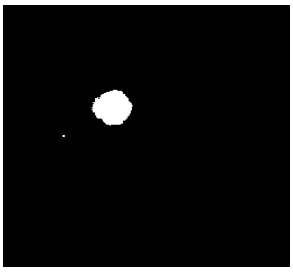
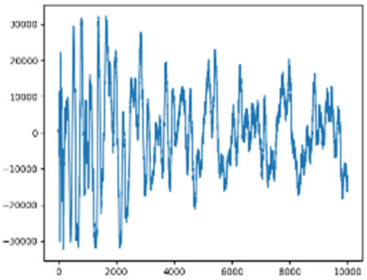
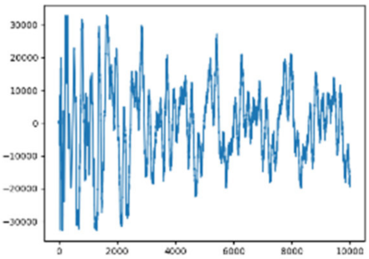
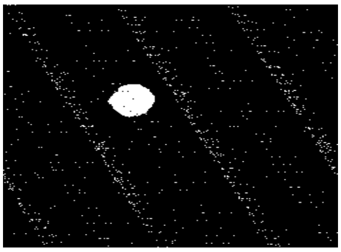
The proposed digital audio watermarking system with SLOA is developed using Python, and the system is configured with the PyCharm software running on the Windows 10 operating system.

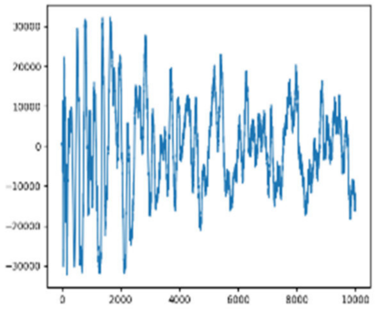
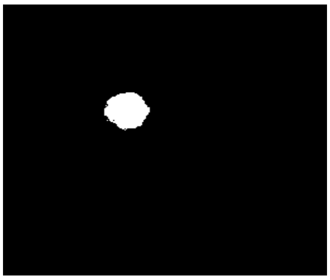
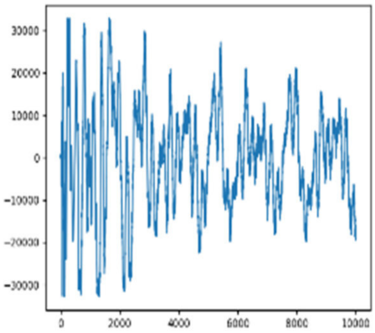
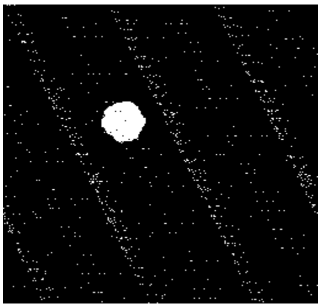
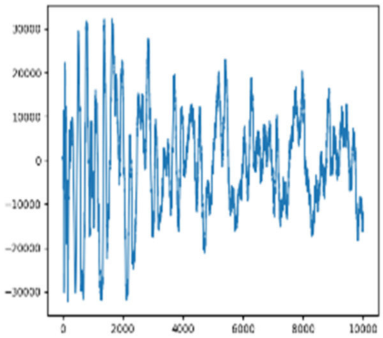
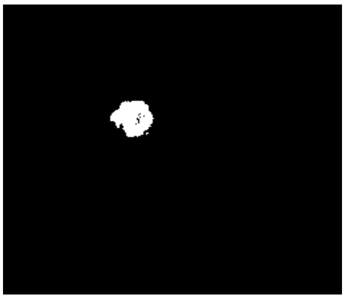
3.5.3 Dataset description: The input image for the brain tumor is sourced from the BraTS database, presenting a multimodal composition with diverse medical scans. Subsequently, experts interpreted the available scans to identify different glioma sub-regions within the BraTS database.

3.5.4 Experimental Result for Proposed Deep CNN with SLOA:

The experimental outcomes for embedding the image into the audio signal are presented in Figure 3.8, showcasing the input signal, the associated embedded image, the recovered image, the original image, and the embedded signal under different noise conditions. The analysis encompasses various types of noise, including salt and pepper noise, Gaussian noise, and random noise. Notably, the proposed method demonstrated robust performance, even in the absence of noise.

Salt and pepper noise	
Input signal	Original image
	
Embedded signal	Recovered image

<table>
<tr><td> </td><td></td></tr>
<tr><td>Input signal</td><td>Original image</td></tr>
<tr><td colspan="2"><b>Gaussian noise</b></td></tr>
<tr><td> </td><td> </td></tr>
<tr><td>Input signal</td><td>Original image</td></tr>
<tr><td> </td><td></td></tr>
<tr><td>Input signal</td><td>Original image</td></tr>
<tr><td colspan="2"><b>Random noise</b></td></tr>
</table>

	
Input signal	Original image
	
Without noise	
	

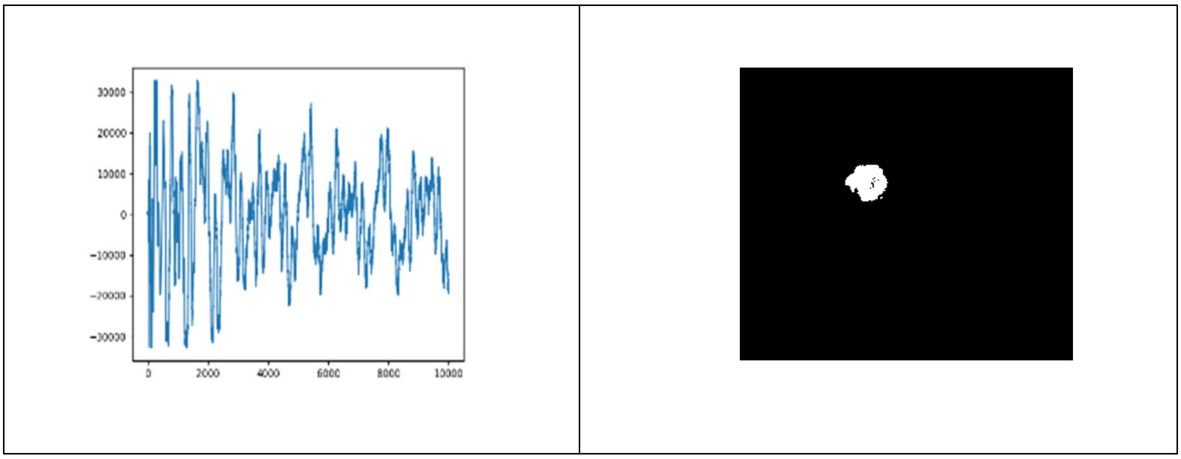


Figure 3.8: Experimental result obtained using the deep CNN with SLOA model

3.5.5 Performance analysis: Basic Neural Network model

The neural network proposed for the watermarking process performs logical calculations to complete the task. This developed neural network comprises two input features and one output feature, with a learning rate set at 0.01. The activation function employed is the Sigmoid function, and the model undergoes training for 100,000 epochs.

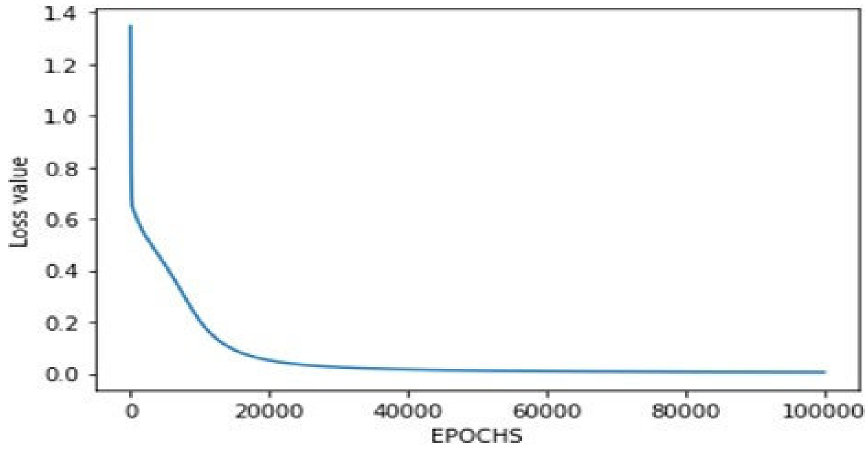


Figure 3.9: Graph representing the model training for 100000 epochs and reduction in loss in NN Model

3.5.6 Achievements and analysis of Deep CNN in audio watermarking:

The effectiveness of the proposed optimization-based digital audio watermarking is delineated through an assessment involving the experimental setup, dataset description, and comparative analysis. The experimental results are compared on the basis of MSE, BER and SNR.

3.5.7 Performance analysis

In this section, the Mean Squared Error (MSE), Bit Error Rate (BER), and Signal-to-Noise Ratio (SNR) of the SLOA optimization are evaluated across five distinct audio signals. The analysis takes into account various noise scenarios, including salt and pepper noise, Gaussian noise, random noise, and a noise-free condition.

3.5.7.1 Analysis based on different signals performance for Image-1:

The performance of SLOA optimization across five different signals under various noise conditions, as assessed by MSE, BER, and SNR, is presented in Table 3.2. In table 3.2, the MSE values for signal 5 are reported for salt and pepper noise, Gaussian noise, random noise, and the noise-free scenario, measuring 0.065, 0.062, 0.059, and 0.056, respectively.

Table 3.2) displays the BER for signal 5, indicating values of 0.081 for salt and pepper noise, 0.081 for Gaussian noise, 0.081 for random noise, and 0.035 without noise.

Table 3.2) showcases the SNR for signal 5 under different noise conditions, with values of 53.422 dB for salt and pepper noise, 54.422 dB for Gaussian noise, 54.422 dB for random noise, and 56.422 dB without noise.

Table 3.2 Analysis Based of Different signals performance with different noise for Image-1

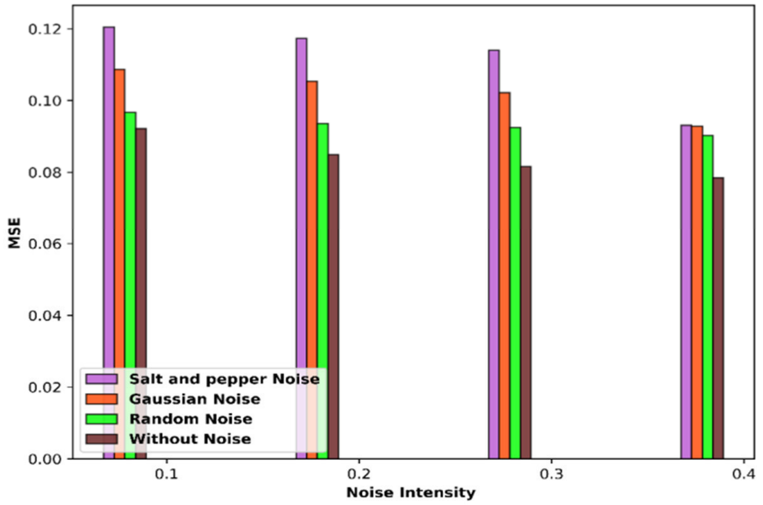
Signals	Salt and pepper noise			Gaussian noise			Random noise			Without noise		
	MSE	BER	SNR (dB)	MSE	BER	SNR (dB)	MSE	BER	SNR (dB)	MSE	BER	SNR (dB)
Signal 1	0.111	0.095	42.363	0.104	0.090	43.363	0.101	0.089	43.363	0.099	0.082	45.363
Signal 2	0.107	0.085	43.378	0.094	0.084	44.378	0.091	0.084	44.378	0.082	0.071	46.378
Signal3	0.097	0.084	45.041	0.083	0.083	46.041	0.081	0.083	46.041	0.080	0.059	48.041
Signal 4	0.086	0.083	50.756	0.075	0.083	51.756	0.073	0.083	51.756	0.070	0.047	53.756
Signal5	0.065	0.081	53.422	0.062	0.081	54.422	0.059	0.081	54.422	0.056	0.035	56.422

3.5.7.2 Performance Analysis for Image-1 based on noise intensity

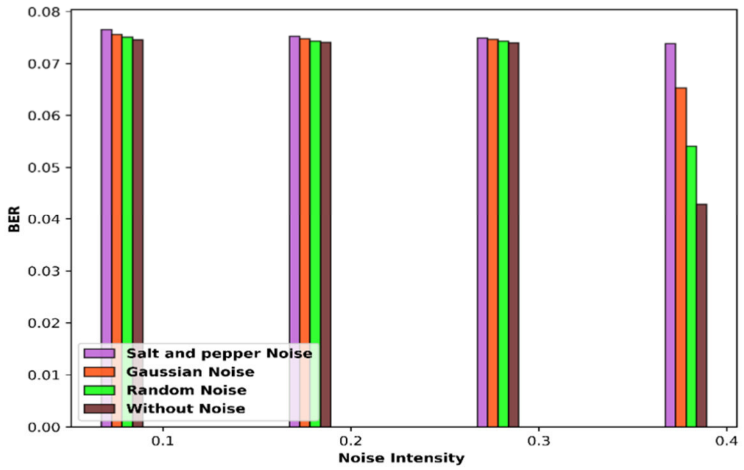
The SLOA optimization performance for various noise intensities using the various noise in terms of the MSE, BER, and SNR are revealed in Figure 3.10. Figure 3.10 a) represents the MSE for both the noise intensity and their corresponding noises. The MSE for the 0.4 noise intensity based on the 3 different noises, and without noise are 0.093, 0.093, 0.090, and 0.078.

Figure 3.10 b) represents the BER for both the noise intensity and their various corresponding noises. The BER for the 0.4 noise intensity based on the 3 different noise, and without noise are and without noise are 0.074, 0.065, 0.054, and 0.043.

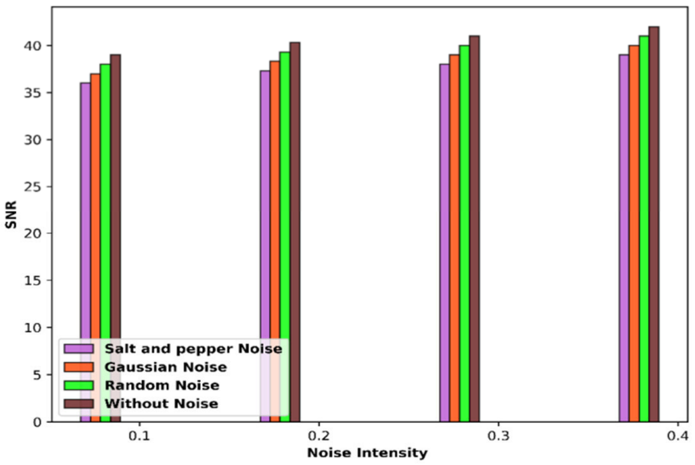
Figure 3.10 c) represents the SNR for both the noise intensity and their various corresponding noises. The SNR for the 0.4 noise intensity based on the 3 different noises, and without noise are 39.001 dB, 40.001 dB, 41.001 dB, and 42.001 dB respectively.



a)



b)



c)

Figure 3.10: Performance analysis for Image-1 a) MSE, b) BER, c) SNR

3.5.7.3 Performance analysis based on different signals for Image-2.

The performance of SLOA optimization across five different signals under various noise conditions, as measured by MSE, BER, and SNR, is presented in Table 3.3. In table 3.3 the MSE values for signal 5 are reported for three different noises and the noise-free scenario, measuring 0.046, 0.042, 0.038, and 0.034, respectively.

Table 3.3) displays the BER for signal 5, indicating values of 0.079 for the three different noises and 0.051 without noise.

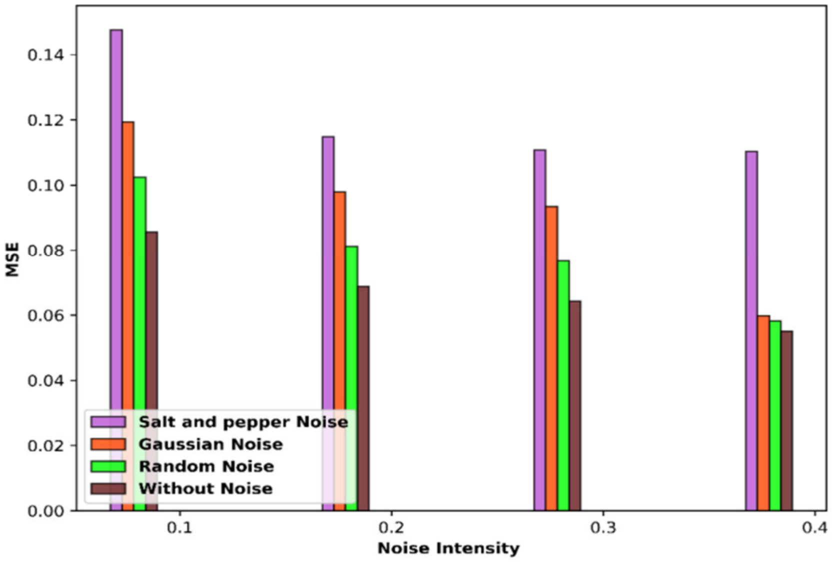
Table 3.3) showcases the SNR for signal 5 under different noise conditions, with values of 57.634 dB for the three different noises and 60.634 dB without noise.

Table 3.3 Analysis Based of Different signals performance with different noise for Image-2

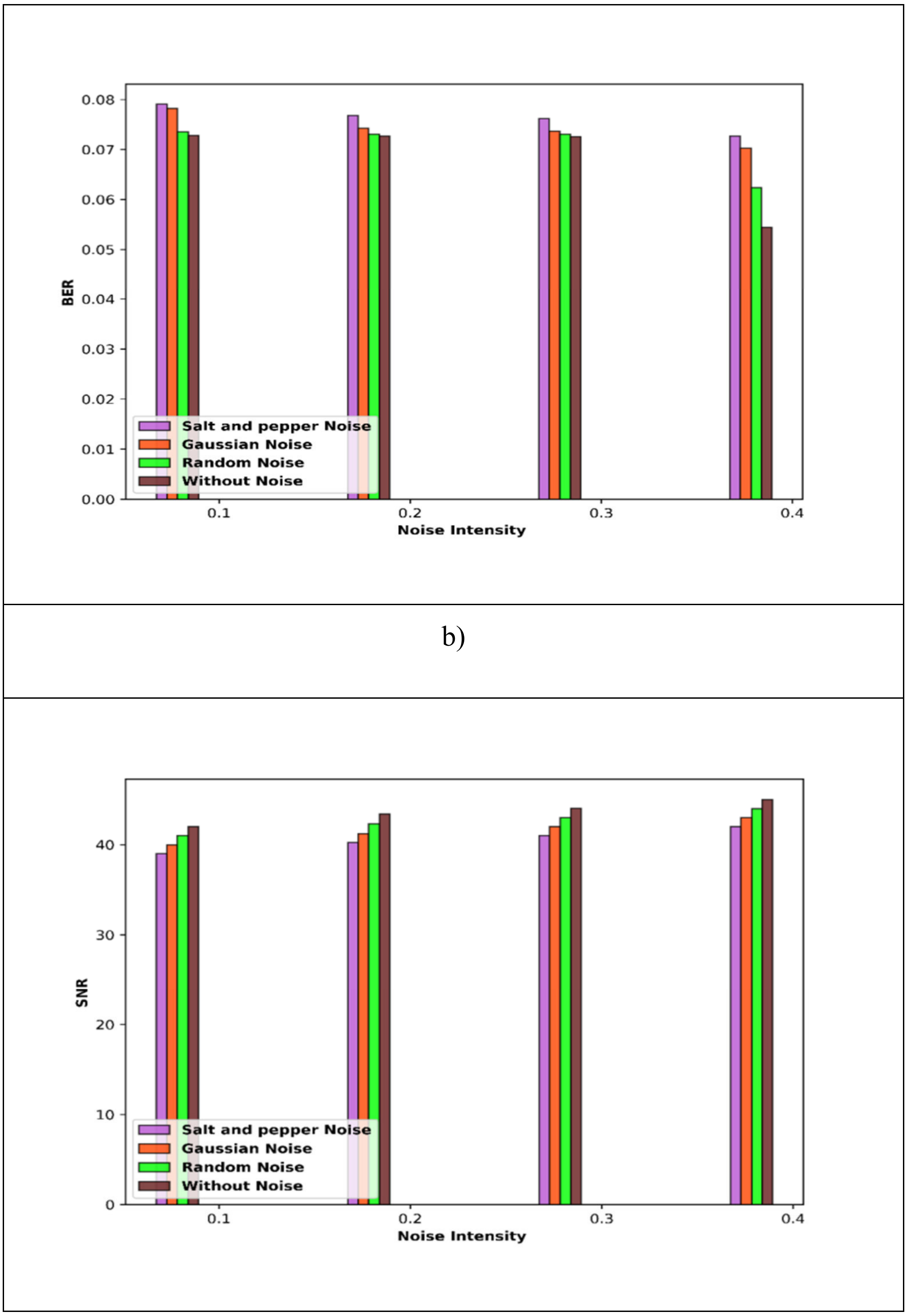
Signals	Salt and pepper noise			Gaussian noise			Random noise			Without noise		
	MSE	BER	SNR (dB)	MSE	BER	SNR (dB)	MSE	BER	SNR (dB)	MSE	BER	SNR (dB)
Signal 1	0.131	0.086	43.045	0.102	0.084	44.045	0.098	0.083	45.045	0.098	0.079	46.045
Signal 2	0.106	0.085	44.067	0.087	0.081	45.067	0.083	0.080	46.067	0.053	0.077	47.067
Signal 3	0.091	0.080	46.062	0.072	0.080	47.062	0.068	0.080	48.062	0.052	0.068	49.062
Signal 4	0.076	0.079	54.134	0.061	0.079	55.134	0.057	0.079	56.134	0.049	0.059	57.134
Signal 5	0.046	0.079	57.634	0.042	0.078	58.634	0.038	0.078	59.634	0.034	0.051	60.634

3.5.7.4 Performance analysis for Image-2 based on noise intensity

The performance of SLOA optimization under various noise intensities, assessed through MSE, BER, and SNR, is depicted in Figure 3.11. In Figure 3.11a), the MSE values for a noise intensity of 0.4 are reported for three different noises and the noise-free scenario, measuring 0.110, 0.060, 0.058, and 0.055, respectively. Figure 3.11b) illustrates the BER for different noise intensities, with values of 0.073, 0.070, 0.062, and 0.054 for a noise intensity of 0.4 with three different noises and without noise. Figure 3.11c) depicts the SNR for various noise intensities, reporting values of 42.001 dB, 43.002 dB, 44.004 dB, and 45.011 dB for noise intensity of 0.4 with three different noises and without noise.



a)



b)

Figure 3.11: Performance analysis for Image-2 a) MSE, b) BER, c) SNR

3.6 Summary:

In conclusion, this research significantly advances the field of digital audio watermarking by proposing a novel deep learning-based system that effectively addresses the limitations of traditional methods. Leveraging the DWT and an optimized DCNN, the proposed model excels in selecting optimal embedding locations, enhancing robustness against various attacks. The meticulous hyper parameter tuning through search location optimization contributes to minimizing errors in the classifier. Experimental results underscore the superior performance of the proposed model, achieving a BER of 0.082, MSE of 0.099, and SNR of 45.363, surpassing existing watermarking models. The research also provides a comprehensive overview of hybrid and novel techniques in digital audio watermarking, emphasizing the efficacy of neural network architectures, particularly the DCNN, in elevating the security and performance of watermark embedding and extraction processes. This work not only demonstrates the feasibility of incorporating deep learning into audio watermarking but also underscores the potential for future advancements in securing digital data.