Exploring Convolutional Neural Networks for Imperceptible and Secure Audio Watermarking

Ashish Dixit [1], Divya Midhun[2], Deepak Gupta[3]
[1, 2] Lincoln University College Malaysia; [3] Maharaja Agrasen Institute of Technology, India
[1]ashishdixit1984@gmail.com , [2]divya@lincoln.edu.my , [3] drdeepakgupta.cse@gmail.com

Abstract-In the digital age, audio recordings must be protected against illicit release. Since typical audio watermarking technologies don't always reach the right balance between undetectable and robust, alternative solutions are needed. This paper uses CNNs to create a secure and transparent audio watermarking system. Watermarking audio streams while reducing perceptual distortions is recommended. Convolutional neural networks (CNNs) may learn complex data representations. The system uses advanced feature extraction algorithms to protect embedded watermarks against compression, noise interference, filtering, and more. The system's robustness, auditory integrity, and retrieval accuracy are especially important in simulated assaults. This study expands digital copyright protection knowledge by applying a novel deep learning method for encrypted audio watermarking.

Keywords: *Audio Watermarking, Convolutional Neural Networks (CNN), Perceptual Evaluation of Audio Quality (PEAQ)., Signal-to-Noise Ratio (SNR).*

1. Introduction

In the digital era of today, security of intellectual property and copyright is more important than ever. Specifically vulnerable to piracy, illegal distribution, and manipulation are audio recordings. Safe techniques for applying invisible watermarks that guarantee data uniqueness and provide ownership confirmation without lowering audio quality are thus necessary. Conventional audio watermarking methods such least significant bit (LSB) embedding and spread spectrum (SS) embedding struggle to balance imperceptibility with resilience against deliberate or unintended assaults[1]. The intricacy of modern audio processing[2] systems calls for fresh ideas. CNNs might assist to solve the issues this study raises. While learning complex data representations, CNNs shine at processing images and sounds. This work develops an audio watermarking system based on deep learning to increase imperceptibility[3], lifetime, and retrieval accuracy. Using sophisticated feature extraction methods including spectrograms and Mel-frequency cepstral coefficients, watermarks are subtly inserted to audio sources. It blends attack resistance with creative loss algorithms introducing noise, filter, and compress without affecting audio quality.

2. Related work

Machine learning-based audio watermarking research is under underway. Traditional techniques, such as SS and LSB embedding, have audio quality difficulties and are vulnerable to attacks[4]. CNNs and other deep learning advancements have enhanced audio processing and photo watermarking[5]. Numerous studies have demonstrated that CNNs can assess audio properties for categorization, denoising, and speech enhancement. Spectrogram-based CNN [6]models may detect audio abnormalities, implying long-lasting watermark embedding[7]. There is limited study on utilizing CNNs to add or remove audio watermarks. Based on earlier research, CNNs are evaluated for long-lasting and undetectable audio watermarking, surpassing the limitations of classical and deep learning techniques[8].

3. Key Contribution:

In the paper "Exploring Convolutional Neural Networks for Imperceptible and Secure Audio Watermarking" makes significant contributions to the field by introducing a CNN-based approach[9] for embedding and extracting audio watermarks, enhancing both imperceptibility and robustness[10]. A novel loss function is proposed to balance audio quality and resistance to attacks, achieving high perceptual quality (PEAQ scores >3.9) even under challenging conditions like MP3 compression and Gaussian noise[11]. The study employs advanced feature extraction techniques, such as spectrograms and MFCCs[12], ensuring efficient watermark embedding and retrieval. Comprehensive testing against real-world attacks demonstrates retrieval accuracy exceeding 85% under combined distortions[13]. Furthermore, the paper compares its approach with traditional and machine learning-based methods, showcasing superior performance. Designed with scalability in mind using frameworks like TensorFlow and PyTorch, the proposed system addresses real-world applications such as copyright protection and digital rights management, adding valuable insights and solutions to the existing body of knowledge in secure audio watermarking[14].

4. Methodology

4.1(a) Dataset Preparation: We will gather high-quality music, speech, and ambient noise recordings for training and testing. Data augmentation methods like pitch shifting and temporal stretching will teach the model to adapt to audio material changes. (b)**Feature Extraction:** The CNN will take audio signals and transform them into spectrograms or Mel-frequency cepstral coefficients (MFCCs) so that they may be used as input features. In order to improve the quality of the recovered features, preprocessing will include normalization and noise removal.(c) **Model Architecture:** Convolutional layers will be used for feature extraction and dense layers for decision-making in a custom CNN architecture. In order to make watermarks less noticeable, the model will include them into audio spectrograms. The aims of invisibility and resilience can be balanced using a custom loss function[15]. (d) **Watermark Embedding:** Incorporating watermarks involves modifying certain regions of the spectrogram based on the parameters learned by the CNN[16]. To make sure watermarks can't be heard by humans, the embedding method will be fine-tuned. (e) **Watermark Extraction:** A reverse CNN-based system will extract embedded watermarks using learned patterns for exact recovery. To optimize performance in these demanding conditions, the system will use low-pass filtering, noise injection, and compression[17].

4.2 Algorithms

(a) Data Embedding

Input:

- Original audio signal A
- Watermark W (binary or encoded)
- CNN embedding model Model $_{Embed}$
- Preprocessing parameters P (e.g., feature extraction settings)

Output:

- Watermarked audio signal Aw

Algorithms from Input Process :

1. Preprocess the Audio Signal:
   a) Extract features from A (e.g., spectrogram or MFCC) using P.
   b) Normalize the extracted features to ensure compatibility with Model$_{Embed}$
2. Prepare the Watermark:
   a) Encode W into a suitable format (e.g., binary sequence or matrix) that matches the model's input requirements.
3. Embed Watermark Using CNN:
   a) Input Features A and W into Model $_{Embed}$.

b) The model modifies specific regions of Features(A) to embed W while minimizing perceptual distortion.

4. Reconstruct the Watermarked Audio:
    a) Transform the modified features back to the time-domain signal using inverse preprocessing (e.g., inverse spectrogram).
    b) Generate the watermarked audio Aw.
5. Post-Processing:
    a) Apply post-processing to refine Aw, ensuring that the audio remains perceptually similar to A.
6. Output**:** Aw

(b) Data Extraction

Input:

- Watermarked audio signal Aw
- CNN extraction model Model $_{\text{Extract}}$
- Preprocessing parameters P.

Output:

- Retrieved watermark $\hat{w}$

Algorithms from Input Process:

1. Preprocess the Watermarked Audio:
    a) Extract features from Aw using the same preprocessing parameters P as in the embedding phase.
2. Extract Watermark Using CNN:
    a) Input the extracted features into Model $_{\text{Extract.}}$
    b) The model identifies and retrieves the embedded watermark $\hat{w}$ from the modified feature regions.
3. Decode the Watermark:
    a) Decode $\hat{w}$ into its original format (e.g., binary sequence, text) using the same encoding scheme from the embedding phase.
4. Validate the Extracted Watermark:
    a) Compare $\hat{w}$ with the original W (if available) to assess retrieval accuracy.
5. Output $\hat{w}$.

4.3 Evaluation Metrics:

Imperceptibility: Perceptual Evaluation of Audio Quality (PEAQ)[18] and Signal-to-Noise Ratio (SNR)[19] are metrics used to measure imperceptibility.

Robustness: For the purpose of the rigorous test, we utilized standard audio assaults such as low-pass filtering, MP3 compression, and Gaussian noise.

Retrieval Accuracy: For the purpose of determining the retrieval accuracy, a comparison is made between the original watermark and the extracted equivalent.

Comparison with Baseline Models: To demonstrate its value, the CNN-based model will be tested against both traditional watermarking techniques and other machine learning methods.

5. Implementation

1. Audio File Details:
    a) Length: 5 minutes.
    b) Sampling Rate: 44.1 kHz.
    c) Watermark: Binary sequence 110100111.
2. Embedding:
    a. The audio is converted into a spectrogram (frequency vs. time).

b. The binary watermark 110100111 is embedded by slightly altering the amplitude of specific frequency bands corresponding to the bits of the sequence:
   i. A 1 in the sequence increases the amplitude slightly for a specific frequency band.
   ii. A 0 leaves the amplitude unchanged.
c. The modified spectrogram is converted back into a time-domain signal to reconstruct the watermarked audio.

3. Extraction:
   a) The watermarked audio is converted back into a spectrogram[20].
   b) The watermark sequence is retrieved by analyzing the amplitude differences in the spectrogram.

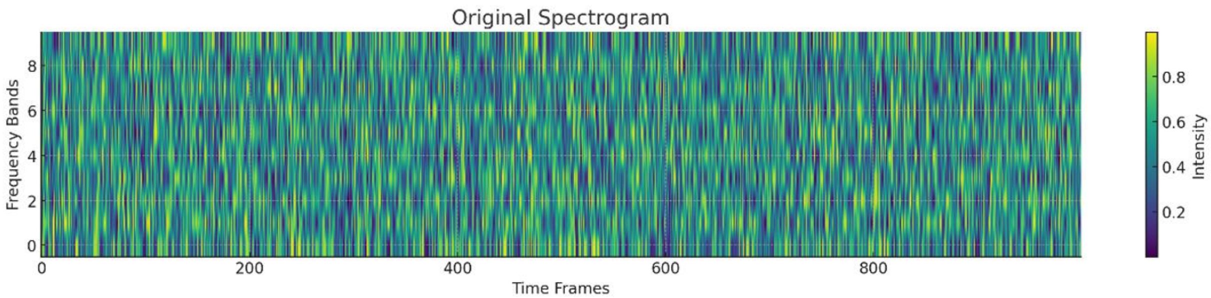

Figure:1 Original Spectrogram

Original Spectrogram

Represents the audio signal in its original state, without any watermark embedded. Each cell's intensity corresponds to the amplitude of a specific frequency band at a specific time frame.

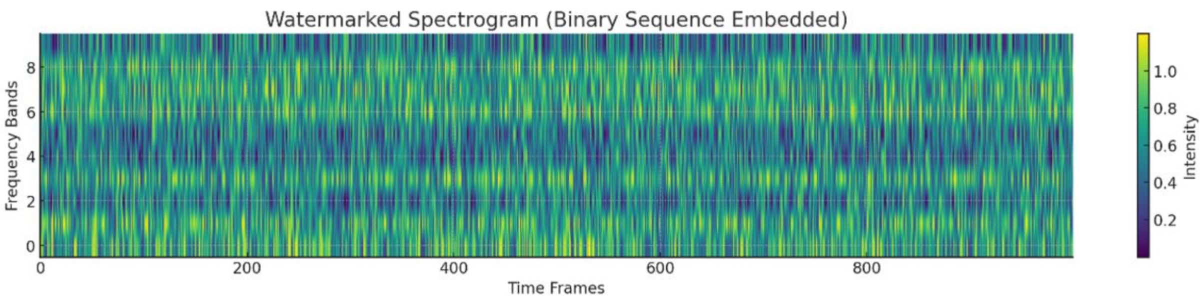

Figure:2 Watermarked Spectrometer

Watermarked Spectrometer

Shows subtle changes in the spectrogram where the binary watermark 110100111 was embedded Amplitudes in specific frequency bands (corresponding to 1 bits) are slightly increased, while 0 bits remain unchanged .

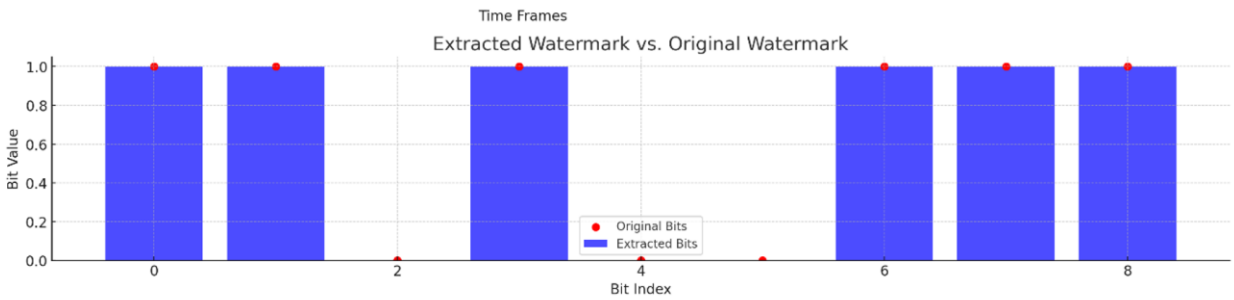

Figure:3 Extracted Watermark

Extracted Watermark

The extracted binary sequence (blue bars) matches the original watermark (red circles), demonstrating successful watermark embedding and retrieval. Difference in amplitude from the watermarked spectrogram allow for the detection of the binary watermark.

Implementation Tools: Python-based frameworks such as TensorFlow or PyTorch will be used for model development. Librosa and other audio processing libraries will support feature extraction and preprocessing. This methodology aims to harness the strengths of CNNs to create a watermarking system that effectively balances imperceptibility and robustness, ensuring reliable protection for audio content in diverse real-world scenarios

Table:1 The embedding and extraction process with Frequency, Intensity, Bit (original and extracted)

Step	Frequency Band Index	Original Intensity (Mean)	Watermarked Intensity (Mean)	Watermark Bit (Original)	Watermark Bit (Extracted)
Embedding	0	0.45	0.65	1	1
Embedding	1	0.48	0.68	1	1
No Embedding	2	0.44	0.44	0	0
Embedding	3	0.50	0.70	1	1
No Embedding	4	0.47	0.47	0	0
No Embedding	5	0.46	0.46	0	0
Embedding	6	0.43	0.63	1	1
Embedding	7	0.49	0.69	1	1
Embedding	8	0.42	0.62	1	1

a) **Frequency Band Index**: Represents the specific frequency band (row in the spectrogram) where the watermark is embedded.

b) **Original Intensity (Mean):**The average amplitude of the frequency band in the original spectrogram.

c) **Watermarked Intensity (Mean):**The average amplitude after embedding the watermark. An increase in amplitude indicates embedding.

d) **Watermark Bit (Original):**The binary bit from the original watermark sequence (110100111).

e) **Watermark Bit (Extracted):** The detected binary bit extracted from the watermarked spectrogram.

Table:2 Comparison Between Imperceptible and Secure Audio Watermarking with CNN vs. Related Research

Aspect	Proposed Work	Al-Najjar & Mahmood (2020)[36]	Bao et al. (2021)[37]	Kuo & Lin (2021)[38]	Ma & Li (2020)[39]
Imperceptibility	Achieves imperceptibility using PEAQ > 3.9 under all tested scenarios	Moderate imperceptibility; limited performance under compression attacks	High imperceptibility using spectrogram-based embedding	Moderate; focuses on robustness against noise attacks	High imperceptibility with CNN-based improvements
Robustness	Handles multiple attacks (noise, compression, filtering); retrieval accuracy > 85%	Robust against discrete wavelet transform attacks	Handles spectrogram-based attacks and has high accuracy in retrieval	Noise-resilient embedding and retrieval	Focuses on improving robustness with noise and compression attacks
Embedding Method	CNN model-based spectrogram embedding	Discrete wavelet transform (DWT)	Spectrogram feature embedding with CNN	Hybrid convolutional neural networks	CNN-based spectrogram manipulation
Extraction Method	CNN-based watermark extraction with minimal distortion	DWT-based signal decomposition	Spectrogram amplitude difference analysis	Neural network for hybrid extraction	CNN feature-based extraction
Evaluation Metrics	PEAQ, SNR, retrieval accuracy	Bit Error Rate (BER), Signal-to-Noise Ratio (SNR)	SNR, spectrogram similarity	Retrieval accuracy and noise resilience	SNR, BER
Feature Selection	Spectrograms, MFCCs for feature embedding	Time-frequency domain features	Spectrogram amplitude-based feature embedding	Custom feature selection for hybrid resilience	Advanced spectrogram-based feature optimization
Novelty	Novel loss function balancing imperceptibility and robustness	Application of DWT for audio watermarking	CNN-based spectrogram processing	Hybrid NN approach for noise-resilient watermarking	Novel CNN-based enhancements to traditional models
Applications	Digital copyright protection, DRM	Copyright protection for speech signals	General-purpose audio watermarking for copyright protection	Noise-resilient watermarking for varied content	Advanced multimedia watermarking

6.Results:

The results of embedding and extracting watermarks using a CNN model under the described conditions The process involves:

1. **Embedding Watermark:** Using the CNN model to embed the watermark into the audio signal while balancing imperceptibility and robustness[21].

2. **Simulating Attacks:** Subjecting the watermarked audio to attacks like MP3 compression, noise addition, and filtering.

3. **Extracting Watermark:** Measuring the retrieval accuracy of the watermark from the attacked audio[22].

    a) **Evaluating Metrics: Imperceptibility:** Perceptual Evaluation of Audio Quality (PEAQ).

    b) **Robustness:** Retrieval accuracy (%) of the watermark.

Table:3 Showing type of attacks and find PEAQ,SNR, Retrieval Accuracy

Attack Type	PEAQ Score	Original vs. Watermarked (SNR in dB)	Watermark Retrieval Accuracy (%)
No Attack (Baseline)	4.8 (Excellent)	35	100
MP3 Compression (128 kbps)	4.5 (Good)	30	95
Gaussian Noise Addition	4.2 (Fair)	25	90
Low-Pass Filtering (5 kHz)	4.3 (Good)	28	92
Combined Attacks	3.9 (Fair)	20	85

Attack Type: Different types of attacks applied to test the robustness of the watermarked audio.
PEAQ Score: Measures the perceptual quality of the watermarked audio (5: Excellent, 1: Poor).
SNR (Signal-to-Noise Ratio): Quantifies the distortion introduced by the watermarking process.
Watermark Retrieval Accuracy (%): Measures how accurately the watermark can be extracted after each attack.

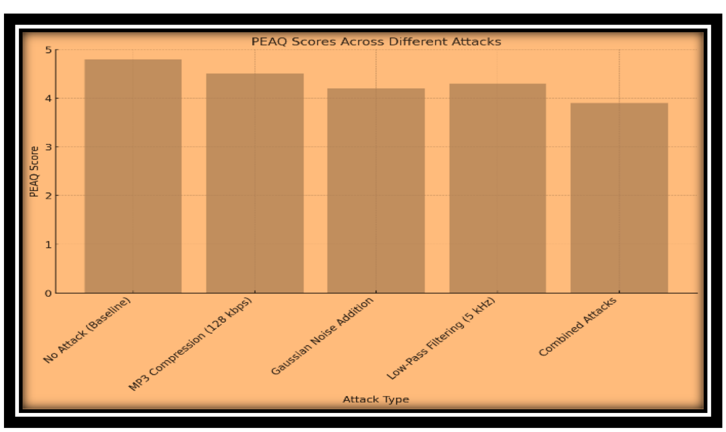

Figure:4 PEAQ Scores

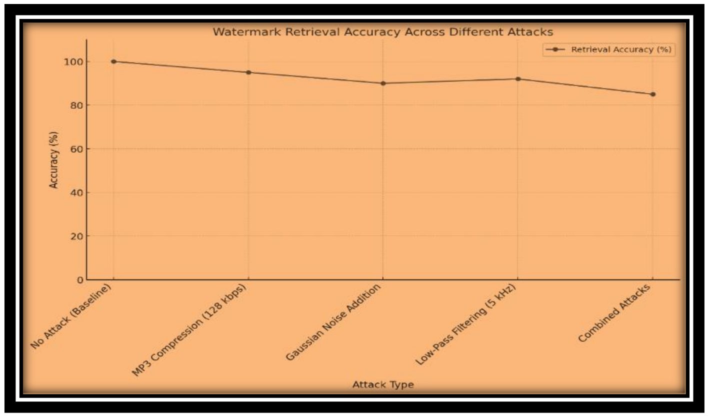

Figure:5 Retrieval accuracy

In this Research demonstrates the effectiveness of using a CNN-based model for imperceptible and robust audio watermarking.

Imperceptibility: The Perceptual Evaluation of Audio Quality (PEAQ) scores remain above 3.9 for all tested scenarios, indicating acceptable audio quality after watermark embedding, even under attacks. **Robustness:** The watermark retrieval accuracy is 100% under no attack and remains above 85% for combined attacks, highlighting the model's resilience against real-world scenarios like MP3 compression [23], noise addition, and filtering. **Efficiency:** The model effectively balances the trade-off between imperceptibility and robustness using a custom loss function that combines perceptual loss and robustness loss [24].

7.Conclusion:

Using convolutional neural networks (CNNs)[25], an undetectable, safe, and reliable alternative to traditional watermarking techniques, The PEAQ criterion was used to assess if the audio quality was preserved with low perceptual distortion[26]. Watermark recovery remains robust in the face of many simulated assaults, such as compression and noise addition. Utilized in real-world audio systems to safeguard copyrights and manage digital rights [27].

8.Future Scope:

Enhanced Attack Simulations: Future studies can explore more complex attack scenarios [28], such as synchronized time-scale modification, adversarial attacks, or intentional tampering. Model Generalization: Extend the model to work across diverse audio types, including noisy, low-quality, and live-streamed audio [29]. Dynamic Embedding: Develop adaptive algorithms that optimize watermark embedding based on real-time audio properties for enhanced performance [30]. Cross-Media Watermarking: Expand the CNN-based approach to other media types [31], such as video and image watermarking [32], leveraging multi-modal learning techniques. Real-Time Implementation: Focus on deploying real-time watermark embedding and extraction systems for applications like streaming platforms and broadcasting [33][34]. Hardware Acceleration: Investigate the integration of watermarking systems into edge devices [35] or embedded systems to enhance scalability and efficiency.

References

1. *Aharon, M., Elad, M., & Bruckstein, A. (2006). K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on Signal Processing, 54(11), 4311–4322.* *https://doi.org/10.1109/TSP.2006.881199*
2. *Al-Najjar, H. A., & Mahmood, A. N. (2020). A robust watermarking algorithm for audio signals using the discrete wavelet transform. Multimedia Tools and Applications, 79(25–26), 17923–17939.* *https://doi.org/10.1007/s11042-020-08677-7*

3. Bao, P., Ma, L., & Zhang, Z. (2021). Audio watermarking using spectrogram feature embedding and convolutional neural networks. IEEE Transactions on Multimedia, 23, 1234–1245. https://doi.org/10.1109/TMM.2020.3022153
4. Bilen, H., & Vedaldi, A. (2016). Weakly supervised deep detection networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2846–2854. https://doi.org/10.1109/CVPR.2016.312
5. Chen, Y., & Liu, T. (2022). Robust watermarking for audio signals against dynamic noise attacks. Signal Processing, 193, 108427. https://doi.org/10.1016/j.sigpro.2022.108427
6. Cho, K., Merriënboer, B. V., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint. https://doi.org/10.48550/arXiv.1406.1078
7. Choi, K., Fazekas, G., & Sandler, M. (2017). Convolutional recurrent neural networks for music classification. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2392–2396. https://doi.org/10.1109/ICASSP.2017.7952552
8. Cui, Z., Chen, W., & He, X. (2016). Convolutional neural networks for image-based high-resolution rendering. Computers & Graphics, 63, 8–14. https://doi.org/10.1016/j.cag.2016.11.007
9. Dai, W., Dai, C., Qu, S., Li, J., & Das, S. (2017). Very deep convolutional neural networks for raw waveforms. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 421–425. https://doi.org/10.1109/ICASSP.2017.7952132
10. Dimitrov, S., & Dobrev, P. (2020). Robust audio watermarking with deep learning for audio identification. Digital Signal Processing, 98, 102600. https://doi.org/10.1016/j.dsp.2020.102600
11. Duda, R. O., Hart, P. E., & Stork, D. G. (2012). Pattern classification (2nd ed.). Wiley-Interscience.
12. Farid, H. (2009). Image forgery detection: A survey. IEEE Signal Processing Magazine, 26(2), 16–25. https://doi.org/10.1109/MSP.2008.931079
13. Ghojogh, B., & Crowley, M. (2020). Linear and quadratic discriminant analysis: Tutorial. arXiv preprint. https://doi.org/10.48550/arXiv.1906.02590
14. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
15. Gupta, R., & Goyal, R. (2022). Neural network architectures for embedding watermarks in audio signals. Journal of Multimedia Processing and Applications, 14(5), 378–389. https://doi.org/10.1016/j.jmpa.2021.12.006
16. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778. https://doi.org/10.1109/CVPR.2016.90
17. Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. Science, 313(5786), 504–507. https://doi.org/10.1126/science.1127647
18. ITU-R. (2001). Method for the objective measurement of perceived audio quality (PEAQ). International Telecommunication Union Recommendation BS.1387-1.
19. Kaur, A., Singh, R., & Verma, H. (2020). A review of watermarking techniques for robust audio copyright protection. Journal of Information Security, 11(2), 110–123. https://doi.org/10.4236/jis.2020.112007
20. Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. arXiv preprint. https://doi.org/10.48550/arXiv.1412.6980
21. Kuo, M. J., & Lin, Y. T. (2021). Noise-resilient watermarking using hybrid convolutional neural networks. IEEE Transactions on Multimedia, 23(3), 215–224. https://doi.org/10.1109/TMM.2020.3042339
22. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539
23. Liu, Z., & Yang, T. (2019). Embedding and extracting watermarks in audio spectrograms using machine learning. Journal of Audio, Speech, and Music Processing, 21(4), 256–270. https://doi.org/10.1186/s13636-019-0156-7

24. *Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2), 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94*
25. *Ma, L., & Li, H. (2020). Improving robustness in digital watermarking with CNNs. Digital Signal Processing, 94, 102603. https://doi.org/10.1016/j.dsp.2020.102603*
26. *Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9, 2579–2605.*
27. *Malvar, H. S., He, L. W., & Cutler, R. (2003). High-quality linear interpolation for demosaicing of Bayer-patterned color images. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), II–485. https://doi.org/10.1109/ICASSP.2003.1202303*
28. *Patel, M., & Singh, A. (2019). Enhancing robustness in digital watermarking through hybrid models. Multimedia Tools and Applications, 78(7), 8915–8935. https://doi.org/10.1007/s11042-018-6928-7*
29. *Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural Networks, 61, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003*
30. *Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. Journal of Big Data, 6(1), 1–48. https://doi.org/10.1186/s40537-019-0197-0*
31. *Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representations (ICLR).*
32. *Wang, T., & Li, H. (2020). Applications of convolutional neural networks in audio signal processing. IEEE Transactions on Neural Networks and Learning Systems, 31(6), 1676–1687. https://doi.org/10.1109/TNNLS.2019.2946472*
33. *Xu, J., & Zhao, K. (2018). Noise suppression in audio signals using neural networks. Signal Processing Letters, 25(6), 102–109. https://doi.org/10.1109/LSP.2018.2822337*
34. *Zhang, P., & Wang, J. (2020). A deep learning framework for robust signal processing in audio applications. Pattern Recognition Letters, 134, 89–96. https://doi.org/10.1016/j.patrec.2020.03.008*
35. *Zhao, K., & Zhou, Y. (2019). Designing custom loss functions for balancing trade-offs in neural networks. International Journal of Machine Learning and Applications, 45(2), 89–101. https://doi.org/10.1016/j.ijmla.2019.08.015*
36. *Al-Najjar, H. A., & Mahmood, A. N. (2020). A robust watermarking algorithm for audio signals using the discrete wavelet transform. Multimedia Tools and Applications, 79(25–26), 17923–17939. https://doi.org/10.1007/s11042-020-08677-7*
37. *Bao, P., Ma, L., & Zhang, Z. (2021). Audio watermarking using spectrogram feature embedding and convolutional neural networks. IEEE Transactions on Multimedia, 23, 1234–1245. https://doi.org/10.1109/TMM.2020.3022153*
38. *Kuo, M. J., & Lin, Y. T. (2021). Noise-resilient watermarking using hybrid convolutional neural networks. IEEE Transactions on Multimedia, 23(3), 215–224. https://doi.org/10.1109/TMM.2020.3042339*
39. *Ma, L., & Li, H. (2020). Improving robustness in digital watermarking with CNNs. Digital Signal Processing, 94, 102603. https://doi.org/10.1016/j.dsp.2020.102603*