Neural Network Based Deep-fake Audio Detection Model

Prof. Kauser Ahmed P.[a], Shaan Chandra[a], Uddipan Sarkar[a], Tiasa Jana[a], Sana Vaidya[a], Anishwar Chakraborty[a]

[a]Vellore Institute of Technology, Vellore, Tamil Nadu 632014, India

Abstract. This study focuses on combating the growing threat of deep-fake voices, where real voices are replaced by synthetic or false ones, through the development of effective detection methods. It explores the use of Recursive Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) and compares their efficacy. The RNN approach involves extracting FFTs and Mel Frequency Coefficients from audio files, converting them into arrays, and feeding them into the network. However, this method yielded a subpar accuracy of 57%. Building on this, the CNN method extends the RNN approach by converting FFTs into spectrograms, preprocessing them into image arrays, and inputting them into the CNN. This led to a significant improvement, with the CNN achieving an accuracy of 95%. By leveraging CNNs and spectrogram representations, this study provides a promising solution for detecting deep-fake voices and mitigating their harmful effects, including misinformation dissemination.

Keywords: Deep-fake voices, Synthetic media, Artificial intelligence (AI), Manipulation, Voice replacement, Deep-fake detection, Threat mitigation, Synthetic voice generation.

1. Introduction

The research objectives include designing and executing a neural network-based system that can effectively differentiate between authentic and manipulated audios. Additionally, the study seeks to integrate a range of techniques, including feature extraction, temporal analysis, and multimodal analysis, all geared towards elevating the precision of deepfake detection. These methods consider critical attributes such as pitch, timbre, and temporal dynamics. Furthermore, the investigation aims to assess the usefulness of spectrograms and the Fast Fourier Transform (FFT) as tools for representing audio data during neural network training, ultimately contributing to improved accuracy in detecting deepfakes.

To achieve our research objectives, we will employ a comprehensive methodology. We will focus on the development of neural networks, designed, and trained to excel in differentiating between genuine and manipulated audios. This stage will involve the utilization of labelled datasets to train our models effectively. Additionally, we will delve into feature extraction techniques, exploring various methods to convert audio data into formats suitable for neural network input. In this regard, we will assess the effectiveness of spectrograms and the utilization of the Fast Fourier Transform (FFT) to create frequency domain representations. Temporal analysis methods will also be implemented to capture dynamic changes within audio data, aiding in the detection of inconsistencies indicative of deepfake content.

This paper presents a holistic approach to deepfake detection by using FFTs and spectrograms as input metrics to streamline computations, while emphasizing the need to strike a balance between accuracy and speed during model evaluation. By contributing insights into the evolving landscape of deepfake detection, this paper aims to make a valuable addition to the field.

1.1. Audio Processing and Feature Extraction

In pursuit of harnessing the full potential of the training dataset, the widely acclaimed Libros library is employed for the crucial task of audio feature extraction. This pivotal step enriches our dataset with essential information by yielding two distinct types of audio features: Fast Fourier Transforms (FFTs) and Spectrograms. These features are instrumental in characterizing and comprehending the auditory content of our audio samples.

Fast Fourier Transforms (FFTs). Fast Fourier Transforms serve as a cornerstone in this feature extraction process. They unveil the intricate frequency components within the audio. This information is invaluable for understanding the underlying spectral characteristics of the audio, providing critical insights into its composition.

Spectrograms. Spectrograms can be narrowed down to a visual representation of the obtained FFTs in a dynamic density-oriented graph.

1.2. Introduction to Fast Fourier Transforms (FFTs)

The Fast Fourier Transform (FFT) is a fundamental and ubiquitous mathematical tool in signal processing, used extensively in various scientific and engineering disciplines. It plays a pivotal role in the decomposition of time-domain signals into their frequency components, facilitating the analysis of complex waveforms and the extraction of critical information from them. In this section, we will provide a comprehensive explanation of FFT, starting from its foundational concepts and building up to its practical applications.

Some fundamental concepts are discussed below -

Time and Frequency Domains. Before delving into FFT, it is essential to comprehend the concept of the time and frequency domains. In the time domain, signals are represented as functions of time (Fig 1.). These signals can be complex and dynamic, such as audio waveforms or physiological data. However, to gain insights into their underlying structures and characteristics, it is often advantageous to transform them into the frequency domain.

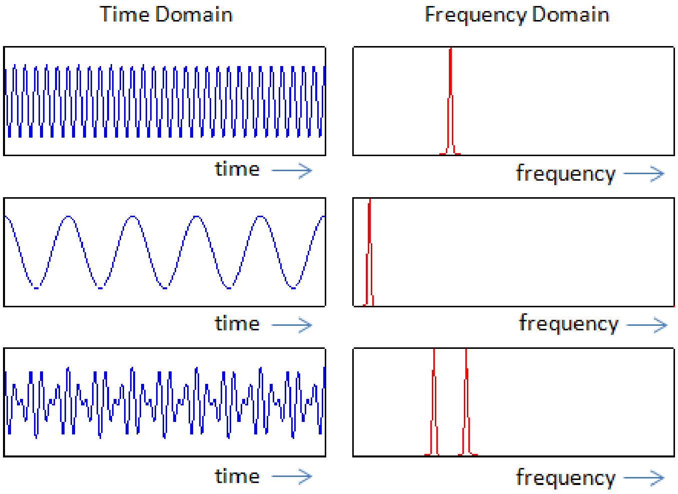

Fig 1. Signals represented as functions of time and frequency.

Decomposition through Fourier Analysis. Fourier analysis is the key to understanding how a complex signal can be decomposed into its constituent sinusoidal components. This process is grounded in the fundamental insight that any periodic waveform can be expressed as a sum of sine and cosine functions of different frequencies and amplitudes. This mathematical principle, originally developed by Jean-Baptiste Joseph Fourier in the 19th century, forms the theoretical foundation of the Fourier Transform.

The Discrete Fourier Transform. The Discrete Fourier Transform (DFT) is a mathematical technique used to convert a discrete, finite-length time-domain signal into its frequency-domain representation (Fig 2.). It provides a mathematical framework for the decomposition of signals into their constituent sinusoidal components. The DFT of a signal x[n], where n represents the discrete time index, is defined mathematically in Eqn. (1) -

$$X(k) \ = \sum_{n=0}^{N-1} x(n)e^{-j2\pi\frac{kn}{N}} \tag{1}$$

Where:

- X[k] is the complex spectrum at frequency bin k.
- x[n] is the discrete time signal.
- N is the length of the signal.

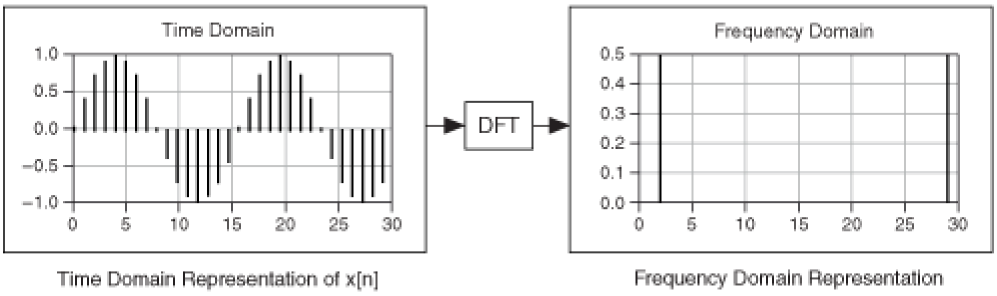

Fig 2. Time and Frequency Domain Representations

Calculation of DFT by its regular definition can be computationally extensive as it requires O(N^2) operations for an N-point signal.

The Fast Fourier Transform. The Fast Fourier Transform (FFT) is an algorithmic technique designed to efficiently compute the DFT of a signal. It was originally introduced by Cooley and Tukey in the 1960s and has since become a cornerstone of signal processing.

The FFT drastically reduces the computational complexity of the DFT, making it possible to compute the DFT of a signal with N points in O(NlogN) operations. This efficiency has revolutionized the field of signal processing, enabling real-time analysis and the processing of vast datasets.

Fast Fourier transform algorithms can typically be categorized into two primary categories: those that involve decimation in time and those that involve decimation in frequency. The Cooley-Tukey FFT algorithm, for instance, starts by reorganizing the input elements in a bit-reversed order and subsequently constructs the output transform (involving decimation in time). The core concept behind this approach is to divide a transform of size N into two smaller transforms of size N/2, a process facilitated by a specific mathematical identity as given in Eqn. (2) -

$$\sum_{n=0}^{N-1} a_n e^{-2\pi ink/N} = \sum_{n=0}^{N/2-1} a_{2n} e^{-2\pi i(2n)k/N} + \sum_{n=0}^{N/2-1} a_{2n+1} e^{-2\pi i(2n+1)k/N} = \sum_{n=0}^{N/2-1} a_n^{even} e^{-2\pi ink/(N/2)} + e^{-2\pi ik/N} \sum_{n=0}^{N/2-1} a_n^{odd} e^{-2\pi ink/(N/2)} \tag{2}$$

This is occasionally referred to as the Danielson-Lanczos lemma. A convenient way to conceptualize this process might be using the Fourier matrix.

The Sande-Tukey algorithm (Stoer and Bulirsch 1980) first transforms, then rearranges the output values (decimation in frequency).

2. Literature Review

Deepfake voices are synthetic media where a person's voice is replaced or impersonated by fake ones, often created using AI techniques. While initially for entertainment, they pose risks like spreading misinformation. One solution is using AI, particularly neural networks, which learn patterns from data and excel at recognizing complex patterns. In deep-fake detection, neural networks can distinguish between real and fake videos by learning subtle differences. They're also powerful for speech recognition, trained on labeled speech samples to identify features and patterns, optimizing performance iteratively.

Detecting deep-fake artificial voices from natural voices is a challenging task, as deepfake technologies are becoming increasingly sophisticated. Some of the most promising approaches to train neural networks include:

- Feature extraction: This involves extracting features from the audio that can be used to distinguish between real and fake voices. For example, neural networks can be trained to identify differences in the pitch, timbre, and loudness of the voice.[1]
- Temporal analysis: This involves analyzing the temporal dynamics of the audio. For example, neural networks can be trained to identify differences in the way that the lips move in real and fake videos.[6]
- Multimodal analysis: This involves combining features from multiple sources, such as audio. This can help to improve the accuracy of detection.[7]

While neural networks are widely used for deepfake detection, they struggle with advanced deep fake voices. Future research should prioritize developing models capable of effectively detecting state-of-the-art deepfakes. Access to diverse datasets is vital for training reliable detection models, yet addressing biases remains a challenge. Data augmentation, employed in this study, helps expand and diversify datasets, enhancing model performance. Combining multiple techniques improves detection accuracy; feature extraction transforms audio data aspects like pitch and frequency for neural network processing. Using spectrograms, which capture both amplitude and frequency information, enriches datasets for more efficient model training. Convolutional layers process these spectrograms in neural networks, and fine-tuning during testing refines the model's ability to distinguish between deep fake and authentic voices.[8]

Convolutional Neural Networks (CNNs) are known for their high complexity due to many variables, posing implementation challenges. Various methods and techniques have emerged to mitigate this complexity issue, including quantization and pruning. However, a novel approach to simplifying CNNs is computation in the Fourier domain. Leveraging the Fast Fourier Transform (FFT) streamlines CNN computations, making them more efficient and faster.[9]

Hence, a more effective approach to detect deepfake audio involves employing Fast Fourier Transforms (FFTs) and directing this transformed data into a dense neural network layer. FFTs, a mathematical technique, enable the decomposition of audio signals into their constituent frequency components. This technique proves invaluable in extracting essential features for deep-fake detection, encompassing factors like voice pitch, frequency, and amplitude.

The Fast Fourier Transform (FFT) will convert the audio signals from the time domain to the frequency domain. This will result in a spectrogram-like representation. [10]

After designing and training the neural networks model, the model will be assessed, and its performance will be evaluated. During evaluation, it is necessary to consider the trade-off between accuracy and speed. A system that is very accurate may be too slow to be practical, while a system that is very fast may not be accurate enough.

Overall, the paper aims to provide a valuable contribution to the field of deep-fake detection.

3. Proposed Methodology

3.1. Data Collection and Dataset Preparation

In the pursuit of advancing language identification systems and enabling groundbreaking research in the realm of speech recognition, our study relies on the harmonious fusion of two meticulously curated datasets. These datasets, each with its distinct attributes, collectively form a potent resource for language and voice-related research.

- Common Languages: This dataset comprises speech recordings obtained from a thoughtfully curated subset of languages sourced from the CommonVoice database. In total, the dataset encompasses 45.1 hours of audio recordings, equating to one hour of speech material available for each of the selected languages. This dataset has been specifically extracted from CommonVoice to serve as a training resource for language identification systems.

- ML Spoken Words: The dataset stands as a substantial and expanding audio collection encompassing spoken words across 50 languages, collectively spoken by a populace exceeding 5 billion individuals. This comprehensive resource serves the dual purpose of facilitating academic research and catering to various commercial applications, particularly in the realms of keyword spotting and spoken term search. Notably, it is made available under the CC-BY 4.0 licensing model.

The dataset itself is voluminous, with an excess of 340,000 distinct keywords, comprising a cumulative total of 23.4 million spoken examples, each lasting one second (equivalent to a cumulative duration exceeding 6,000 hours). Its utility spans a wide spectrum, with applications extending from the integration of voice-activated consumer devices to enhancing call center automation.

3.2. Data Processing

Extraction of FFTs with respect to Hamming Windows. In the rigorous data analysis process, we adopt a method that involves dividing the waveform of each audio sample into distinct time frames. This segmentation technique is carried out using the Hamming Window method, a well-established approach for splitting the audio signal into manageable segments.

Following the segmentation, we apply the Fast Fourier Transform (FFT) to each of these individual time windows of the audio waveform. The FFT operation unveils the frequency content of each time window, offering crucial insights into the dynamic changes occurring within the audio signal as it evolves over time.

The outcome of this process yields a series of FFT curves, each corresponding to a specific time window. These FFT curves are graphically represented, with each distinct color serving as a visual indicator of a different time window (Fig 3.). This visual representation allows us to observe and analyze how the frequency components of the audio signal change over time, providing a dynamic view of the audio's spectral characteristics throughout its duration.

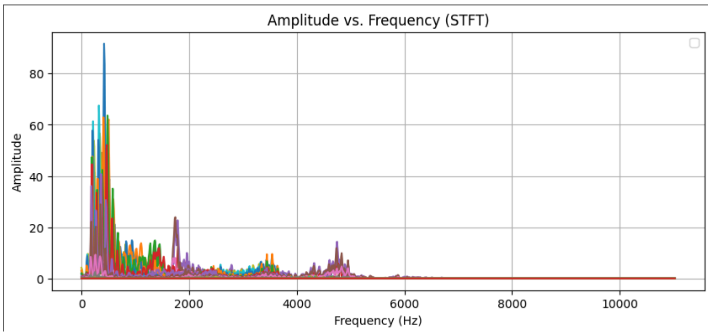

Fig 3. Amplitude vs. Frequency (STFT)

Spectrograms. Spectrograms are visual representations of the frequency content of a signal over time. They are commonly used in signal processing, audio analysis, and speech recognition to analyze and understand the characteristics of a signal.

To create a spectrogram, you typically start with a time-domain signal, such as an audio waveform. The process involves dividing the signal into short, overlapping segments. Each segment is then transformed into the frequency domain using a mathematical tool called the Fourier Transform.

The most common method for computing the Fourier Transform of short segments of a signal is called the Short-Time Fourier Transform (STFT). In STFT, the signal segment is multiplied by a window function (such as Hamming or Hann window) to minimize spectral leakage. Then, the Fourier Transform is applied to each windowed segment to obtain its frequency content.

Once the frequency content of each segment is computed, it is plotted over time to create the spectrogram. The x-axis represents time, the y-axis represents frequency, and the intensity or color of each point in the plot corresponds to the magnitude or power of the frequency component at that time and frequency.

In our project, the following is an example of the obtained spectrograms.

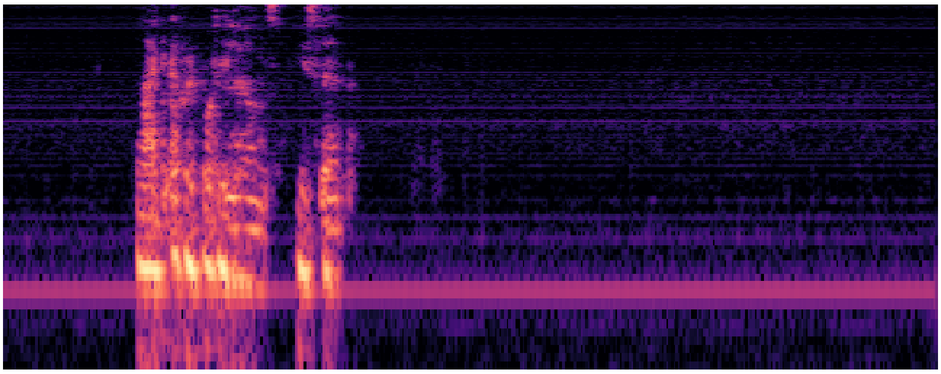

Fig 4. Representation of the obtained spectrograms. (X-axis: time, Y-axis: frequency). Intensity of each point corresponds to the magnitude of the frequency component at that point.

3.3. Model Planning

After extracting the concatenated feature set of FFTs and MFCCs along with the input vector of the spectrograms, we have 2 routes to follow, either choosing FFTs and MFFCs or choosing the spectrograms as our key input feature in our neural network.

Both individual approaches require different types of neural networks. A Recurrent Neural Network (RNN) will provide much more value if proceed with FFTs and MFCCs while a Convolution Neural Network will aid us more if spectrograms are chosen as we would be dealing with images at that point in time.

Approach A (FFTs). We will be using a recurrent neural network (RNN) for this approach.

Recurrent Neural Networks (RNNs) are a type of neural network designed for sequence data, like time series or text. Unlike feedforward networks, they have connections that loop back on themselves, allowing them to maintain a memory of past inputs. This memory enables RNNs to process sequences of inputs, making them useful for tasks like language modeling, speech recognition, and time series prediction. The difference between RNNs and traditional feed-forward neural networks is the factor of recurrence.

The following is the model architecture that is being followed in this approach:

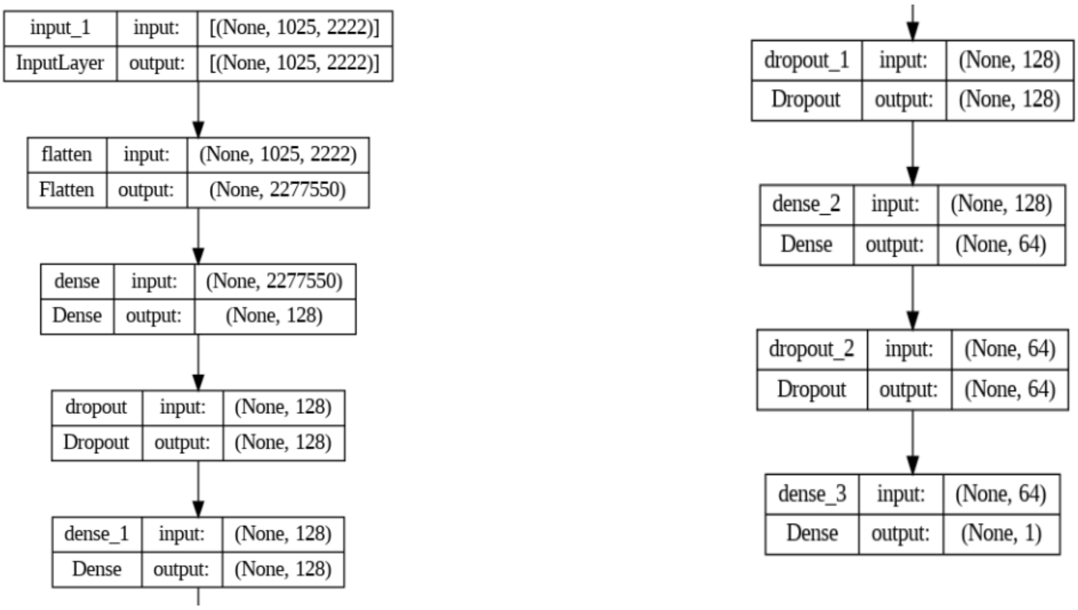
input_1 | input: | [(None, 1025, 2222)]
InputLayer | output: | [(None, 1025, 2222)]
flatten | input: | (None, 1025, 2222)
Flatten | output: | (None, 2277550)
dense | input: | (None, 2277550)
Dense | output: | (None, 128)
dropout | input: | (None, 128)
Dropout | output: | (None, 128)
dense_1 | input: | (None, 128)
Dense | output: | (None, 128)
dropout_1 | input: | (None, 128)
Dropout | output: | (None, 128)
dense_2 | input: | (None, 128)
Dense | output: | (None, 64)
dropout_2 | input: | (None, 64)
Dropout | output: | (None, 64)
dense_3 | input: | (None, 64)
Dense | output: | (None, 1)

Fig 5. RNN Model Architecture

Upon training the model with respect to our training data, an accuracy of 60% was yielded which was clearly not satisfactory and thus, we decided to proceed with another approach where we used spectrograms to train the model.

Approach B (Spectrograms). This approach would rather be called an extension of the first approach instead of an alternative route as spectrograms in the very core are graphical representations of the procured FFTs.

For this approach, we used a convolution neural network.

Convolutional Neural Networks (CNNs) are a type of deep learning model specifically designed for processing structured grid-like data, such as images. They consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers.

Convolutional layers apply filters to input data, extracting features through convolutions. Pooling layers reduce the spatial dimensions of the feature maps, helping to make the network more efficient. Fully connected layers integrate the extracted features to make predictions.

The following is the model architecture that we used:

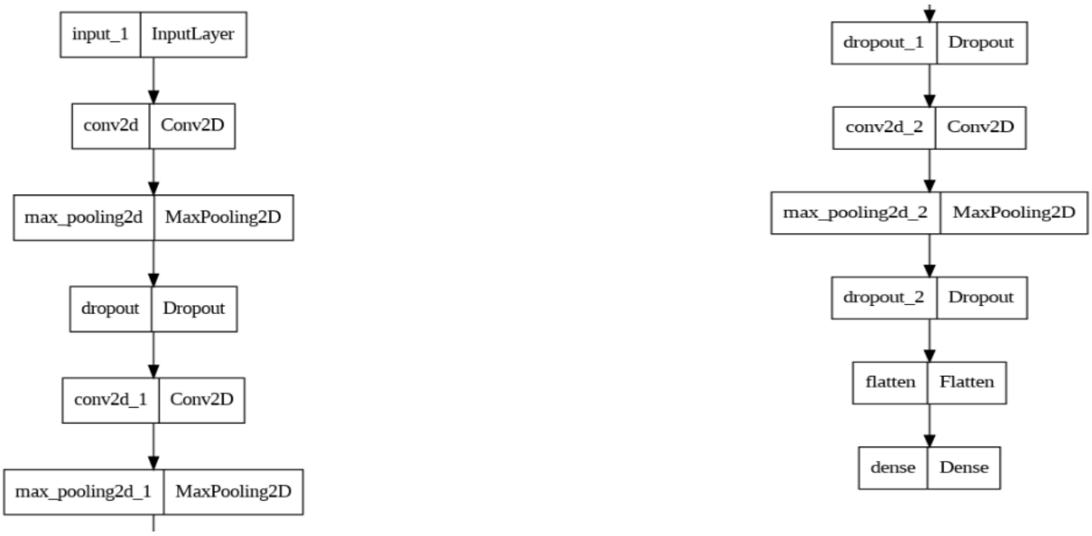

Fig 6. Spectrograms Model Architecture

This approach led to a surprisingly high training accuracy of around 95%.

Along with that in the testing set, it almost predicted a majority of the chunk of the sample set correctly.

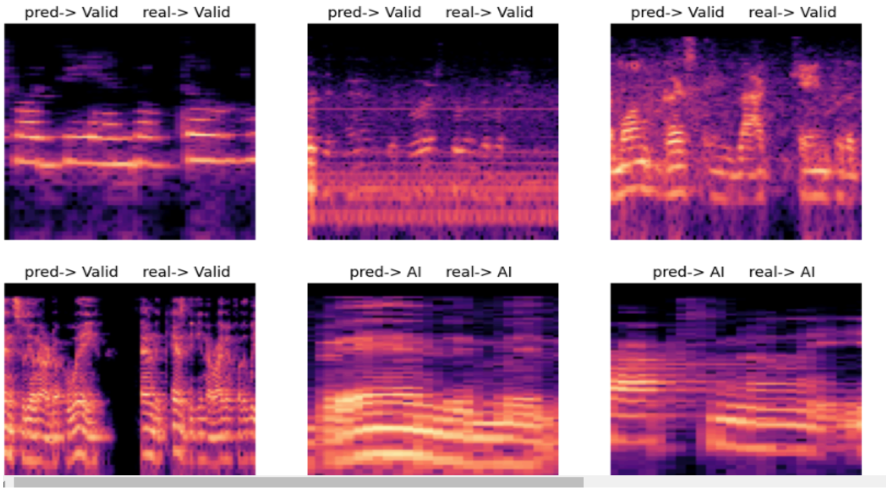

Fig 7. Visual Representation of Accuracy in Prediction

4. Conclusion

This research paper dives deep into the realm of distinguishing authentic voices from artificial or manipulated ones, a task of paramount importance in the age of digital deception. Throughout our investigation, we have not only comprehended the technical intricacies of neural networks but also their practical applications in voice recognition systems. The following is the comparison in between the 2 discussed approaches:

Table 1. Accuracy Comparison between the two Approaches.

Approach	Accuracy
FFTs and RNN	57%
Spectrograms and CNN	95%

As technology continues to advance, the knowledge gleaned from this research serves as a robust foundation for further developments in voice analysis and deception detection. Neural networks, with their ability to discern subtle patterns and nonlinear relationships, are poised to play a pivotal role in safeguarding digital communication from malicious manipulations.

Deepfake audio detection has the potential to serve as a crucial tool in addressing many real-world challenges, such as – fraud prevention, audio authentication, audio evidence verification, media forensics, and so on. This technology can play a big role in preserving authenticity in evidence and media and fostering trust among users and consumers.

References

1. Childer, D.G.: "The Matlab Speech Processing and Synthesis Toolbox." Photocopy Edition, Tsinghua University Press, Beijing, 45-51 (2004).
2. Saxena, R., Shobe, J., and McNaughton, B.: "Learning in deep neural networks and brains with similarity-weighted interleaved learning." Proceedings of the National Academy of Sciences, vol. 119 pp. 15-20 (2022).
3. Singh, M., and Verma, K.: "Speech Recognition Using Neural Networks." Journal of Advances in Information Technology, vol. 2, pp. 108–110, (2011).
4. Kumar, K., and Chaturvedi, K.: "An Audio Classification Approach using Feature extraction neural network classification Approach.", pp. 1–6 (2020).
5. Bahmaninezhad, F. et al.: "A Comprehensive Study of Speech Separation: Spectrogram vs Waveform Separation.", pp. 4574–4578 (2019).
6. Sebyakin, A., Soloviev, V., and Zolotaryuk, A.: "Spatio-Temporal Deepfake Detection with Deep Neural Networks.", pp. 78–94 (2021).
7. Salvi, D. et al.: "A Robust Approach to Multimodal Deepfake Detection." Journal of Imaging, vol. 9, p. 122, (2023).
8. Richards, M., Varshini, E., Diviya, N., Prakash, P., Palanichamy, K., and S. A.: "Deep Fake Face Detection using Convolutional Neural Networks.", pp. 1–5 (2023).
9. Al Smadi, K., Al Issa, H., Trrad, I., and Al Smadi, P.-T.: "Artificial Intelligence for Speech Recognition Based on Neural Networks." Journal of Signal and Information Processing, vol. 06, pp. 66–72, (2015).
10. Lu, J., Wu, Q., Hexun, J., Fu, S., Tang, M., and Lu, C.: "Efficient Timing/Frequency Synchronization Based on Sparse Fast Fourier Transform (S-FFT)." Journal of Lightwave Technology, vol. 05, pp. 1–1, (2019).
11. Weisstein, E. W.: "Fast Fourier Transform.", https://mathworld.wolfram.com/FastFourierTransform.html, last accessed 2024/03/15.
12. Weisstein, E. W.: "Discrete Fourier Transform.", https://mathworld.wolfram.com/DiscreteFourierTransform.html, last accessed 2024/03/15
13. Oberst, U.: "The Fast Fourier Transform." SIAM Journal on Control and Optimization, vol. 46, pp. 496–540, (2007).
14. Ozan, Ö.: "The Discrete Fourier Transform (DFT) and its Relation to the Fourier Transform." vol. 23, pp. 50-63 (2003).
15. Lenssen, N., and Needell, D.: "An Introduction to Fourier Analysis with Applications to Music." Journal of Humanistic Mathematics, vol. 4, pp. 72–91, (2014).
16. Grossi, E., and Buscema, M.: "Introduction to Artificial Neural Networks." European Journal of Gastroenterology & Hepatology, vol. 19, pp. 1046–1054, (2008).
17. Bishop, C.: "Neural Networks and Their Applications." Review of Scientific Instruments, vol. 65, pp. 1803–1832, (1994).
18. Drew, P., and Monson, J.: "Artificial Neural Networks." Surgery, vol. 127, pp. 3–11, (2000).
19. Schmidhuber, J.: "Deep Learning in Neural Networks: An Overview." Neural Networks, vol. 61, pp. 15-19, (2014).
20. Sarker, I.: "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications, and Research Directions." SN Computer Science, vol. 2, pp. 45-67, (2021).
21. Bhavsar, H., and Ganatra, A.: "A Comparative Study of Training Algorithms for Supervised Machine Learning." International Journal of Soft Computing and Engineering (IJSCE), vol. 2, pp. 34-56, (2012).
22. Sekhar, C., and Meghana, P.: "A Study on Backpropagation in Artificial Neural Networks." Asia-Pacific Journal of Neural Networks and Its Applications, vol. 4, pp. 21–28, (2020).
23. Rao, D.: "Fuzzy Neural Networks." IETE Journal of Research (44), 227–236 (2015).
24. Murata, N., Yoshizawa, S., and Amari, S.-I.: "Learning Curves, Model Selection and Complexity of Neural Networks." In: Neural Information Processing System, vol. 5, pp. 607-614, Morgan Kaufmann Publishers (1993).

25. Ahmed, Wajeeha and Chaudhary, Areeshia and Naqvi, Gulfraz, "Role of Artificial Neural Networks in AI.", vol. 20, pp. 3365-3373 (2023).