RESEARCH ARTICLE

RSP Science Hub

International Research Journal on Advanced Science Hub
2582-4376
www.rspsciencehub.com
Vol. 06, Issue 07 July

Check for updates

http://dx.doi.org/10.47392/IRJASH.2024.028

AI Formed Audio and Human Audio Detection

Prof. K. S. Warke[1], Siddhi Choughule[2], Ketki Dandgavale[3], Anjali Mundhe[4]

[1]Assistant Professor, Computer Engineering Department, Bharati Vidyapeeth's College of Engineering for Women, Pune, India.

[2,3,4]Student, Computer Engineering Department, Bharati Vidyapeeth's College of Engineering for Women, Pune, India.

Emails: *kanchan.warke@bharatividyapeeth.edu[1], siddhi.choughule2002@gmail.com[2], ketki.dandgavale@gmail.com[3], mundheanjali15@gmail.com[4]*

Article history

Received: 30 June 2024
Accepted: 21 July 2024
Published: 27 July 2024

Keywords:
Audio detection, Common Voice dataset, Mel-Frequency Cepstral Coefficients, Convolutional Neural Network (CNN), Support Vector Machine (SVM).

Abstract

Our project, "AI Formed Audio and Human Audio Detection," addresses the limitations of current fake audio detection methods by developing an automated end-to-end solution. We leverage a convolutional neural network (CNN) framework to efficiently detect human audio using speech waveforms and acoustic features like MFCCs, which extract high-level representations and consider prosody differences between genuine and fake speech. We utilize the Common Voice dataset from Kaggle for authentic human voice samples, and the pyttsx3 library to convert sentences from the Flickr8k.txt file into male and female synthetic voices. Feature selection and extraction techniques focused on MFCCs ensure robust feature representation, and the dataset is standardized using a Standard Scaler to enhance model performance. Both CNN and the Support Vector Machine (SVM) models were used for classification, with CNN model outperforming the SVM in accuracy. Prioritizing user-friendliness and accessibility, we provide an interactive user interface that accepts audio in various formats, such as WAV and MP3. Our approach, combining automated feature selection, MFCC-based feature extraction, CNN and SVM modelling, and an intuitive interface, accurately detects AI formed audio and human audio, helping to safeguard against misinformation and privacy violations while ensuring accessibility for a broader audience.

1. Introduction

Recently The evolution of audio technology has revolutionized communication and media, yet it has also introduced new challenges in verifying the authenticity of audio recordings. Historically, audio verification relied on manual analysis and subjective judgment, processes prone to error and inefficiency. With the advent of artificial intelligence (AI), particularly in speech synthesis, the landscape of audio authentication has fundamentally shifted. AI algorithms can now generate highly realistic synthetic voices or audios that are increasingly hard to distinguish from genuine human speech. This technological advancement has significant implications across various sectors, including media, cybersecurity, and law enforcement. The ability to create convincing fake audio raises concerns about misinformation, privacy violations, and the potential for identity

fraud. These risks underscore the urgent need for automated detection systems capable of accurately discerning between AI formed audio and human audio. The "AI Formed Audio and Human Audio Detection" project addresses these challenges by leveraging advanced machine learning (ML) techniques, particularly convolutional neural networks (CNNs), to enhance audio verification capabilities. By analysing speech waveforms and extracting acoustic features like as Mel-Frequency Cepstral Coefficients (MFCCs), our system aims to capture subtle nuances that distinguish genuine human speech from synthetic counterparts. This paper explores the historical context and evolution of audio authentication, highlighting the shift from manual methods to automated systems driven by AI. It discusses the limitations of existing approaches and proposes a comprehensive solution that integrates data-driven methodologies with user-friendly interfaces. By providing a detailed examination of our project's methodology and outcomes, we demonstrate how CNNs and MFCCs can significantly increase the accuracy and reliability of audio detection systems. In summary, the integration of AI in audio synthesis presents both opportunities and challenges. This introduction sets the stage for exploring how advancements in machine learning can mitigate the risks associated with AI formed audio, ensuring trust and security in digital communications and media integrity. [1]

2. Literature Survey

The literature on detecting AI formed audio and deepfake voices showcases diverse methodologies and advancements aimed at addressing the ethical and security challenges posed by synthetic audio. Bird and Lotfi (2023) introduced the DEEP-VOICE dataset, focusing on real-time detection of Artificial Intelligence (AI) formed audio speech using statistical analysis of temporal audio or voices features. [2] Their study emphasizes the effectiveness of Extreme Gradient Boosting (XGBoost) models, achieving an impressive 99.3% classification accuracy with rapid processing capabilities, crucial for preventing misuse in identity theft and social engineering. Lim et al. (2022) explored explainable deep learning techniques for deepfake voice detection, employing methods like Deep Taylor and layer-wise relevance propagation (LRP) on CNN and CNN-LSTM models. They highlighted the interpretability of these models in distinguishing deepfake characteristics from genuine audio, crucial for non-expert users in understanding decision-making processes. Hossan et al. (2010) proposed a novel MFCC feature extraction method based on distributed Discrete Cosine Transform (DCT-II), comparing it with conventional techniques using Gaussian Mixture Model (GMM) classifiers. Their study contributes to enhancing the performance of speaker verification systems by improving feature extraction techniques. Sharifuddin et al. (2020) compared CNNs and SVMs for voice recognition in an intelligent wheelchair application, demonstrating CNNs' superior accuracy in distinguishing between voice commands compared to SVMs, despite the latter's faster processing times. Their research underscores the trade-offs between accuracy and computational efficiency in real-world applications. Liu et al. (2021) focused on identifying fake stereo audio using SVM and CNN models, developing algorithms capable of effectively detecting manipulated audio content through robust feature extraction and classification techniques. Their work contributes to enhancing audio forensics capabilities in detecting and mitigating the spread of fake audio content. Hamza et al. (2022) investigated deepfake audio detection using MFCC features and machine learning models, highlighting SVM's effectiveness in detecting different datasets of synthetic audio, underscoring its applicability across various scenarios and datasets. Wang et al. (2022) presented a fully automated end-to-end fake voice detection system using a wav2vec pre-trained model and light-DARTS architecture. Their approach achieves exceptional performance on the ASVspoof 2019 LA dataset, leveraging advanced deep learning techniques to optimize neural architectures for accurate and efficient detection of fake audio. Rana et al. (2022) performed a systematic literature review on deepfake detection, categorizing methodologies into deep learning-based, classical machine learning-based, statistical, and blockchain-based techniques. They conclude that deep learning (DL) approaches generally outperform other methods, highlighting ongoing advancements and challenges in combating deepfake technologies. Lunagaria and Parekh (2020) discussed the implications and risks of deepfake audio,

emphasizing the role of Python and deep learning in developing accurate models for distinguishing between real and fake audio. Their study underscores the need for robust detection mechanisms to mitigate potential societal and security risks associated with deepfake technologies. [3-5]

3. Method

3.1 Proposed System

Designing a system architecture for AI Formed audio and human audio detection involves several components working together to analyze and classify audio signals accurately. The system's functioning is divided into five phases: Data Collection, Audio Processing, Feature Extraction, Training Models, and Classification. [6]

3.2 Data Collection

The data collection process for the AI formed audio and human audio detection system involves curating a comprehensive dataset consisting of over 25,000 audio files each of human and AI formed audio, totaling 5.3 GB. This dataset forms the foundation for training the model.

3.2.1 Human Audio Dataset

The human audio files are sourced from the Kaggle dataset "Common Voice," which includes audio data in MP3 format. This dataset provides a rich variety of speakers' demographics, including age groups (Teens to Nineties), gender (Male, Female, Other), and accents (e.g., US English, Australian English, Indian English). The audio data is organized into folders based on corresponding CSV files, facilitating easy access and management. This structure ensures that the diverse range of accents and demographics can be systematically incorporated into the training process. The Common Voice dataset is publicly available at [Common Voice Dataset] (https://www.kaggle.com/datasets/mozillaorg/common-voice).

3.2.2 AI Formed Audio Dataset

The AI formed audio files are created using the pyttsx3 library. Sentences for generating these audio files are sourced from the "Flick8k.token.txt" file, which contains over 12,000 sentences from a GitHub project named MUTT. Using pyttsx3, these sentences are converted into audio in both male and female voices, resulting in more than 24,000 AI formed audio files. This approach ensures a balanced representation of genders in the AI formed audio dataset, mirroring the diversity found in the human audio dataset. The "Flick8k.token.txt" file is available at [Flick8k.token.txt] (https://github.com/text-machine-lab/MUTT/blob/master/data/flickr/Flickr8k.token.txt).

3.2.3 Dataset Labeling

The entire dataset of 25,000 human-formed audio and 25,000 AI formed audio files is meticulously labeled to distinguish between human and AI formed audio content. Proper labeling is critical for the supervised learning algorithms employed in the system, as it allows the model to learn the distinguishing features of human and AI formed audio effectively. [7-11]

3.3 Preprocessing

This phase involves applying pre-processing techniques to the raw audio data. A key technique used is Mel-frequency cepstral coefficient (MFCC) extraction, where 12 coefficients are extracted from each audio file. Additionally, four other relevant features are extracted from the audio files. These features, along with the pre-processed MFCC features, are stored for further processing. Standard scaling is applied to standardize features by cancelling the mean and scaling to unit variance, rescaling values to a scope between 0 and 1. This centers the data around the mean and scales it to standard deviation of 1, which is useful when the distribution of data is unfamiliar or not Gaussian.

3.3.1 Feature Extraction Using MFCC

After pre-processing, MFCC feature extraction is applied to the pre-processed audio data. The audio is segmented into short frames, and a series of processing steps, including Fourier transforms, Mel filter banks, and cepstral analysis, are performed audio to compute the MFCCs. These extracted MFCC features are stored for use in model training. Other features extracted include Chroma STFT, which computes the short-time Fourier transform and maps it to 12 pitch classes, Spectral Centroid, which measures the centre of mass of the spectrum, Spectral Bandwidth, which measures the width of spectrum, Spectral Roll off, which identifies the frequency next to which a stated percentage of the entire spectral energy lies, and Zero Crossing Rate, which measures the rate of sign variations in the signal.

3.4 Training Models

The classification phase employs machine learning

algorithms to categorize audio into classes. The system uses Convolutional Neural Networks (CNN) and Support Vector Machines (SVM) for this purpose.

3.4.1 Convolutional Neural Networks (CNN)

CNNs are deep learning models commonly used for image classification, object recognition, and tasks involving structured grid-like data, such as images and time-series data. CNNs apply filters (kernels) across the input data to extract hierarchical features. They perform convolutions, pooling operations, and nonlinear activation functions to learn increasingly abstract representations of features present in the input. This hierarchical feature learning captures patterns at various levels of abstraction, from easy edges and textures to complex object parts and shapes. CNNs are well-suited for image-related functions due to their capacity to spontaneously learn relevant information from raw data. They have shown remarkable performance in tasks like as audio classification, object detection, and semantic segmentation. CNNs significantly reduce need for manual feature engineering as they learn hierarchical representations directly from the data.

- **Activation Function:** An activation function, such as Rectified Linear Unit (ReLU), is applied after as in every convolutional layer to introduce non-linearity, granting the network to learn complex relationships within the data. ReLU helps mitigate the vanishing gradient problem and permits faster training equated to other activation functions like sigmoid or tanh.
- **SoftMax:** The SoftMax activation function is utilized in the output layer for multi-class classification problems. It transforms raw output scores (logits) into probabilities, providing a probability distribution over all possible classes.
- **Dense (Fully Connected) Layer:** Dense layers, also known as fully connected layers, receive input from all neurons in the past layer. In audio CNNs, dense layers typically follow convolutional layers and learn to combine extracted features for classification. They use activation functions like ReLU to introduce non-linearity and learn complex relationships in the data.
- **Output Layer:** The output layer of a CNN for audio classification tasks typically uses SoftMax activation for multi-class classification. SoftMax converts the final layer's raw output into class probabilities. The predicted class is determined by choosing the class with highest probability.

3.4.2 Support Vector Machines (SVM)

The SVM model includes data loading and preprocessing, model training, and audio feature extraction and prediction. Data loading involves importing necessary libraries and loading a CSV file named "Features_final.csv" into a Pandas DataFrame. Label encoding transforms categorical labels into numerical values for machine learning algorithms. Feature scaling standardizes the feature set to improve model performance. An instance of SVR (Support Vector Regressor) is initialized and trained using the standardized feature set. The trained SVM model is saved for future use, allowing it to be reused without retraining.

3.5 Audio Feature Extraction and Prediction

A function named get(file_name) is defined to perform audio feature extraction and prediction. It loads an audio file and extracts feature such as chroma_stft, spectral centroid, spectral bandwidth, spectral rolloff, zero crossing rate, and MFCCs. These features are computed and stored as their mean values. The saved SVM model is loaded, and the extracted features are fed into the SVM model to make a prediction. The predicted output is decoded to map the numerical value to a corresponding class label, such as "AI Formed audio" or "Human Voice." This proposed system provides a comprehensive approach to detecting AI formed audio and human audio, utilizing CNN and SVM models for accurate classification and leveraging advanced feature extraction techniques to Enhance Performance. Figure 1 shows System Architecture.

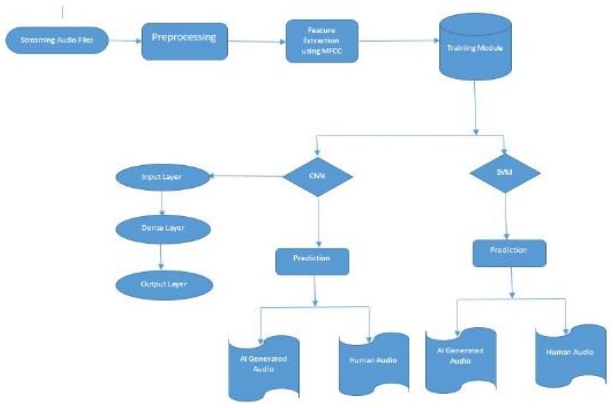

Figure 1 System Architecture

4. Algorithms

4.1 CNN

4.1.1 Data Pre-Processing

Load the dataset containing audio features from "Features_final.csv" using Pandas. Extract the target variable (genre_list) and encode it using LabelEncoder to convert categorical labels into numerical format (0 and 1 in this case). Standardize the input features (X) using StandardScaler to ensure all features have mean of 0 and a standard deviation of 1, which helps in the training the model efficiently. Figure 2 shows CNN Architecture.

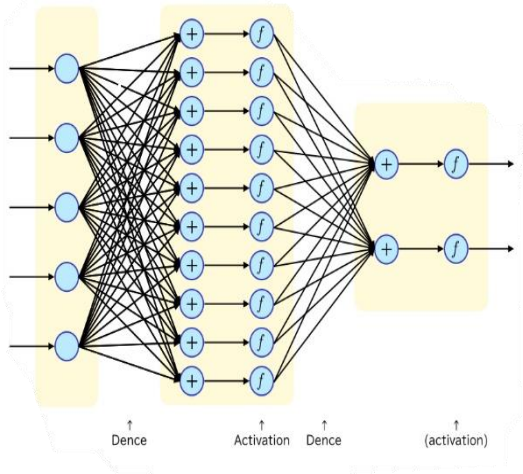

Figure 2 CNN Architecture

4.1.2 Model Creation

Initialize a Sequential model using Keras, high-level neural networks API. Add layers to the model: Input Layer: Configure the input layer to accept the pre-processed features with an input shape corresponding to the number of features in the dataset. Dense Layers: Add multiple dense layers with different numbers of neurons (1024, 512, 256, 128, 64, 32, 16, 8, 4) and 'relu' activation function, which introduces non-linearity into the model and learns high-level representations from input features. Output Layer: Add an output layer with 2 neurons (corresponding to the two categories: AI formed audio and human-formed audio) and 'softmax' activation function, which outputs probabilities for as in every class.

4.1.3 Model Compilation

Compile the model using the Adam optimizer, which is efficient for training deep learning models. Choose 'sparse_categorical_crossentropy' as the loss function since it's suitable for multi-class classification tasks. Specify 'accuracy' as the metric to monitor during training, which measures the model's performance.

4.1.4 Training

Set the number of epochs to 25, signifying the number of times the model will be trained on the whole data-set. Fit the model to the preprocessed input data (X) and encoded object labels (y) using the specified number of epochs. During training, the model adapts its internal parameters (weights and biases) to diminish the loss function and improve accuracy.

4.1.5 Model Saving

Save the trained model ('NN.h5') to a file for future use or deployment. Figure 3 Shows MFCC Feature.

4.2 MFCCs Feature Extraction

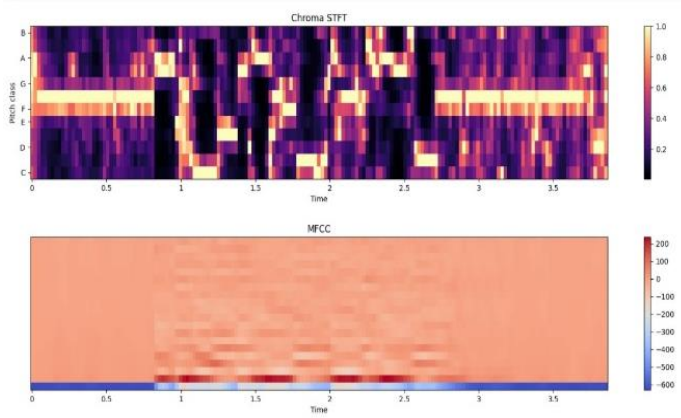

Figure 3 MFCCs Feature Representation

4.2.1 Initialization

Begin by importing crucial libraries such as librosa for advanced audio processing, pandas for data management, numpy for numerical computations, matplotlib for graphical representation, os for file operations, PIL for image processing, pathlib for path manipulation, csv for file handling, keras for deep learning tasks, and warnings for managing system alerts. Define the structure of the CSV file by specifying the header, comprising features like chroma_stft, spectral_centroid, spectral_bandwidth, spectral_rolloff, zero_crossing_rate, and MFCCs, to ensure organized data storage.

4.2.2 Feature Extraction Loop

Establish a systematic loop to traverse through each category of audio files and each individual audio file within those categories, demonstrating an efficient and structured approach to data processing. Utilize the sophisticated capabilities of librosa to load each audio file and extract

pertinent features such as chroma_stft, spectral_centroid, spectral_bandwidth, spectral_rolloff, zero_crossing_rate, and MFCCs, showcasing comprehensive analysis of audio characteristics. Employ statistical techniques to compute the mean value of each feature set, encapsulating the essence of the audio segment and facilitating meaningful data representation. Construct a meticulously crafted string variable ('to_append') encapsulating the extracted features along with the corresponding filename and category label, showcasing a meticulous approach to data organization and annotation.

1. Chroma STFT:

Chroma Short-Time Fourier Transform (STFT) is a technique that combines the traditional STFT with pitch class analysis to capture the harmonic and melodic content of audio signals.

$$X(t,k) = \sum_{n=-\infty}^{\infty} \left\{ x(n) \cdot w(n-t) \cdot e^{\left(-\frac{j2\pi kn}{N}\right)} \right\}$$

X(t,k) is the STFT of frame t and frequency bin
x(n) is the input voice signal.
w(n) is the window function.
R is the hop size.
N is the number of points in the FFT.
j is the imaginary unit.

$$p(k) = k \bmod 12$$

p(k) is the pitch class of frequency bin k.

$$C(t,p) = \sum_{k \in P(p)} \{ \; |X(t,k)| \; \}$$

C(t, p) is the Chroma vector component for frame t and pitch class. P(p) is the set of frequency bins corresponding to pitch class p.

2. Spectral Centroid:

The spectral centroid is an amount used in signal processing and music analysis to indicate where the center of mass of the spectrum is located. It is often perceived as the "brightness" of a voice. Mathematically, it is defined as the weighted mean of the frequencies existed in the signal, with their magnitudes as weights.

$$\text{Spectral centroid} = \frac{\sum_K f(k) \cdot |X(k)|}{\sum_K |X(k)|}$$

f(k) is the frequency at bin k.
X(k) is the magnitude of the STFT at frequency bin k.

3. Spectral Bandwidth

Spectral bandwidth is an amount of the spread of frequencies in this way the spectral centroid. It presents an indication of the range of frequencies present in a signal and can be thought of as the "width" of the spectrum.

$$\text{Spectral Bandwidth} = \sqrt{\frac{\sum_k [f(k) - \mu]^2 \; |X(k)|}{\sum_k |X(k)|}}$$

f(k) is the frequency at bin k
X(k) is the magnitude of the STFT at frequency bin k
μ is the spectral centroid, which is calculated as

$$\mu = \frac{\sum_k f(k) . |X(k)|}{\sum_k |X(k)|}$$

4. Spectral Rolloff

Spectral roll off is an amount used to define the shape of the spectrum of an audio signal. It indicates the frequency beneath which a specified percentage (generally 85% or 95%) of the whole spectral energy is present. This amount helps in distinguishing between harmonic and noisy sounds; harmonic sounds tend to have lower rolloff values, while noisy sounds have higher rolloff values. Calculate the Total Spectral Energy: Sum the magnitudes of all frequency bins.

$$\sum_{total} = \sum_k |X(k)|$$

Determine the Threshold Energy**:** Calculate the specified percentage of the total spectral energy. For instance, for 85% rolloff, the threshold energy would be:

$$\sum_{threshold} = 0.85 \sum_{total}$$

5. Find the Roll-Off Frequency:

Identify the frequency bin krolloff where the cumulative sum of magnitudes first exceeds the threshold energy.

$$\sum_{k=0}^{k_{rolloff}} |X(k)| \geq \sum_{threshold}$$

The corresponding frequency 〖f(k〗 _rolloff) is the spectral rolloff frequency.

In summary, the spectral rolloff frequency 〖f(k〗 _rolloff) is defined as:

$$f_{rolloff} = f(k_{rolloff})$$

Such that

$$\sum_{k=0}^{k_{rolloff}} |X(k)| \geq 0.85 . \sum_k |X(k)|$$

6. Zero Crossing Rate:

The Zero Crossing Rate (ZCR) is simple yet effective feature used in signal processing to characterize aspects of the waveform's shape. It measures the rate at which the signal diverse its hint, which corresponds to the number of times the waveform crosses the horizontal axis (zero amplitude).

$$ZCR = \frac{1}{N-1}\sum_{n=4}^{N-1} I\left[x(n)\cdot x(n-1) < 0\right]$$

N is the length of the signal (number of samples). x(n) denotes the amplitude of the signal at time index n I(·) is the indicator function, which give outcome 1 if its argument is true and 0 otherwise.

7. MFCC (Mel-Frequency Cepstral Coefficients):

Mel-Frequency Cepstral Coefficients (MFCCs) are extensively used features in audio signal processing, especially in speech and music analysis. They are derived from the power spectrum of the sound signal but are processed to better represent how humans perceive sound.

- **Pre-emphasis:** Optionally, the signal may undergo pre-emphasis to amplify higher frequencies that contribute more to the overall spectral shape. Where, α\alphaα is a pre-emphasis coefficient typically around 0.97.

  $x'(n)=x(n)-\alpha\cdot x(n-1)$

- **Frame Blocking:** The signal is splited into overlying frames of generally 20-30 ms, with a 50% overlap between consecutive frames.
- **Windowing:** Each frame is multiplied by a window function (e.g., Hamming window) to cut down spectral leakage.

  x_w(n) = x′ (n)−w(n)

- **Fast Fourier Transform (FFT):** Compute the Discrete Fourier Transform (DFT) of each windowed frame to obtain magnitude spectrum.

  X(k) = FFT(x_w(n))

- **Mel Filterbank:** Apply a Mel filterbank to the magnitude spectrum. The Mel scale is noncognitive scale of pitches that approximates the human auditory system's feedback more closely than linear frequency bands. where S_(m) is the power spectrum of the mmm-th Mel frequency band, |⟦X(k)|⟧^2 is the squared magnitude of the k-th FFT coefficient, and H_m (k) is the Mel filterbank weight for the m-th band at frequency bin k.

  $S_m = \sum_{k=0}^{N-1} | X(k) |^2 \cdot H_m(k)$

- **Logarithm:** Take the logarithm of the Mel filterbank energies to compress dynamic range and mimic human hearing sensitivity to loudness.

  $M_m = \log (S_m)$

- **Discrete Cosine Transform (DCT):** Apply the Discrete Cosine Transform to decorrelate the Mel frequency cepstral coefficients and extract the utmost informative features. where are the MFCCs, M is the number of Mel filters, and α(l) is a normalization factor.

  $C_l = \sum_{m=0}^{M-1} \alpha(l) \cdot M_m$

- **MFCCs:** The resulting coefficients C_l (typically 12-13 coefficients) serve as the short-term power spectrum of the sound in a compact form suitable for different voices processing tasks such as speech recognition, speaker identification, and music genre classification.

4.3 Compose for CSV

Execute a seamless procedure to write the meticulously extracted data from each audio file into the CSV file, ensuring meticulous data integrity and organization. Employ a systematic row-by-row approach to data storage, ensuring each row encapsulates a unique audio segment with its meticulously extracted features and corresponding label, facilitating streamlined data analysis and interpretation.

5. Result And Discussion

The "AI formed audio and Human Audio Detection" project has successfully developed a fully automated end-to-end solution to address the challenges posed by human and fake audio recordings. This innovative system represents a significant advancement in audio analysis, particularly in combating misinformation, identity theft, and privacy violations linked to AI formed audio content. Unlike traditional fake voice recognition systems that depend heavily on manual network parameter adjustments and expert experience, our solution offers a more efficient and automated approach. This significantly reduces potential human error and enhances overall accuracy. At the heart of our method is a meticulously designed Convolutional Neural Network (CNN) framework that leverages

speech waveforms and extracts relevant acoustic features as in every Mel-frequency cepstral coefficients (MFCCs). This method captures intricate prosodic differences between genuine and fake speech, enabling the model to make highly accurate classifications. By integrating feature selection and extraction techniques focused on MFCCs, we have improved the system's performance and adaptability, ensuring robust feature representation for effective audio analysis. Furthermore, the project incorporates a Support Vector Machine (SVM) module that complements the CNN model's capabilities by efficiently classifying audio samples as either AI formed audio or human voice. This integration of diverse machine learning techniques not only increases classification accuracy but also improves system's versatility in handling various types of audio content. The system's real-time detection and classification capabilities underscore its efficacy, demonstrating its ability to categorize audio samples into distinct classes based on learned patterns and features. User experience has been a paramount consideration, reflected in the development of an intuitive user interface. This interface facilitates seamless interaction, allowing users to input audio in various formats and providing real-time visualization of detection results. Additionally, the system offers customizable parameters, enabling users to tailor the analysis to their specific needs. The visualization and interpretation of detection outcomes are presented in a visually compelling manner, aiding users in understanding the system's decisions and enhancing the overall interpretability of detection results. Figure 4,5 Shows the output.

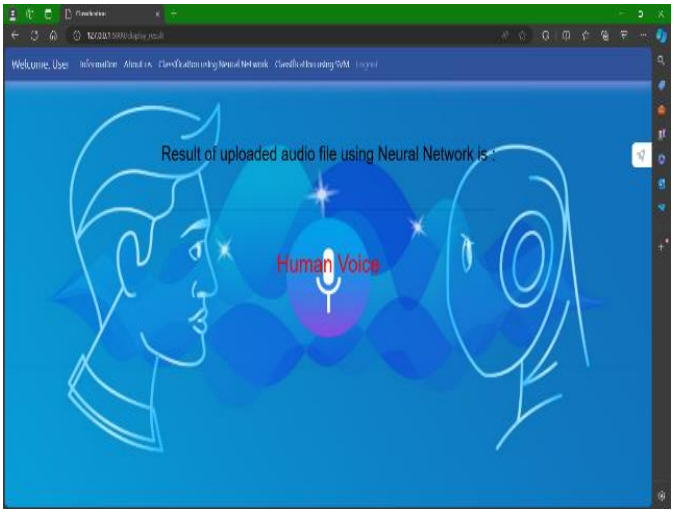

Figure 4 Output as Human Voice for Given Input

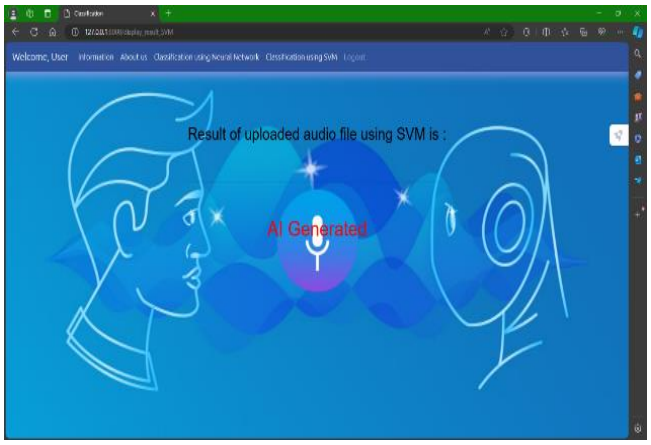

Figure 5 Output as AI Generated Voice for Given Input

Conclusion

"AI formed audio and Human Audio Detection" represents a significant advancement in audio authentication. Utilizing cutting-edge CNN and SVM technologies, along with meticulous data preprocessing, the system effectively distinguishes genuine human speech from AI formed audio. The user-friendly interface ensures accessibility for all users, while the use of the Common Voice dataset and pyttsx3 library promotes inclusivity. Focusing on Mel-Frequency Cepstral Coefficients (MFCCs) enhances model reliability. This comprehensive, end-to-end solution addresses the growing issue of counterfeit audio, reinforcing authenticity and trust in digital communications.

References

[1]. Galyashina, Elena & Nikishin, Vladimir. (2021). “AI Generated Fake Audio as a New Threat to Information Security: Legal and Forensic Aspects” 17-21. 10.5220/0010616700003170.

[2]. Mukhwana, Clinton & Omondi, Evans & Ambale, Diana & Akinyi, Macrine & Mukhwana, Kelvin & Omollo, Richard & Chege, Ben. (2023). “Kenoobi Humanlike Voices: Revolutionizing Audio Content Creation with AI-Generated Human-Like Voices in 140+ Languages Prepared by Kenoobi AI” 10.13140/RG.2.2.10968.47362.

[3]. Mindner, Lorenz & Schlippe, Tim & Schaaff, Kristina. (2023). “Classification of Human- and AI-Generated Texts: Investigating Features for ChatGPT” 10.1007/978- 981-99-7947-9_12.

[4]. Denny, Paul & Khosravi, Hassan & Hellas, Arto & Leinonen, Juho & Sarsa, Sami. (2023).

"Can We Trust AI-Generated Educational Content? Comparative Analysis of Human and AI-Generated Learning Resources" 10.48550/arXiv.2306.10509.

[5]. Dai, Nancy & Lee, Jiyoung & Kim, Ji Won. (2023). "AI vs. Human Voices: How Delivery Source and Narrative Format Influence the Effectiveness of Persuasion Messages" International Journal of Human-Computer Interaction. 10.1080/10447318.2023.2288734.

[6]. J. Bird and A. Lotfi, "Real-time Detection of AI-Generated Speech for DeepFake Voice Conversion," , v1, 2023 arXiv:2308.12734v1 .

[7]. Lim S.-Y., Chae D.-K., and Lee S.-C., "Detecting Deepfake Voice Using Explainable Deep Learning Techniques," Applied Sciences, vol. 12, no. 8, p. 3926, 2022. doi: 10.3390/app12083926.

[8]. M. Hossan, S. Memon, and M. Gregory, "A novel approach for MFCC feature extraction," in Proc. Int. Conf. Signal Process. Commun. Syst., Dec. 2010, pp. 1-5, doi: 10.1109/ICSPCS.2010.5709752.

[9]. M. Sharifuddin, S. Nordin, and A. Ali, "Comparison of CNNs and SVM for voice control wheelchair," IAES International Journal of Artificial Intelligence (IJ-AI), vol. 9, pp. 387- 393, 2020. doi: 10.11591/ijai.v9.i3.pp387-393.

[10]. Wang, Chenglong & Yi, Jiangyan & Tao, Jianhua & Sun, Haiyang & Chen, Xun & Tian, Zhengkun & Ma, Haoxin & Fan, Cunhang & Fu, Ruibo. (2022). "Fully Automated End-to-End Fake Audio Detection." 27-33. 10.1145/3552466.3556530.

[11]. B. Alsaify, H. Abu Arja, B. Maayah, M. Altaweel, R. Alazrai, and M. Daoud, "Voice Based Human Identification using Machine Learning," in Proc. 2022 Int. Conf. Inf. Commun. Sci. (ICICS), 2022, pp. 205-208, doi: 10.1109/ICICS55353.2022.9811154.