A PROJECT REPORT ON

A Novel Approach to Audio Deep Fake Detection using CNN and Bi-LSTM

SUBMITTED TO THE
VISHWAKARMA INSTITUTE OF INFORMATION TECHNOLOGY, PUNE
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

BACHELOR OF TECHNOLOGY (AI & DS)

SUBMITTED BY

Vivek Shinde	Exam No : 22110036
Rohan Jagtap	Exam No : 22111304
Rohan Sonawane	Exam No : 22111303
Suyash Yeolekar	Exam No : 22111297

Under Guidance of
Prof. Dr. Ratna Patil

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE

Affiliated To

Savitribai Phule Pune University, Pune

BRACT'S
VISHWAKARMA INSTITUTE OF INFORMATION TECHNOLOGY , PUNE
2024 -2025

CERTIFICATE

This is to certify that the project report entitled

“ A Novel Approach to Audio Deep Fake Detection using CNN and Bi-LSTM”

Submitted by

Vivek Shinde	Exam No :22110036
Rohan Jagtap	Exam No :22111304
Rohan Sonawane	Exam No :22111303
Suyash Yeolekar	Exam No :22111297

is a bonafide student of this institute and the work has been carried out by them under the supervision of **Prof. Dr. Ratna Patil** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of **Bachelor of Technology** (AI & DS).

Prof. Dr. Ratna Patil
Project Guide
Department of AI & DS

Prof. Santosh Kumar
Head,
Department of AI & DS

Prof. Dr. Vivek Deshpande
Director, VIIT, Pune

Date :

Examiners : 1. 2.

Place : Pune

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who have contributed to the successful completion of this project on "A Novel Approach to Audio Deep Fake Detection using CNN and Bi-LSTM" This endeavor has been an incredible journey, and I owe my thanks to various individuals and resources.

First and foremost, I extend my deepest appreciation to my project guide Ratna Patil, whose expertise, guidance, and unwavering support have been invaluable throughout the entire duration of this project. Their insights and constructive feedback have played a pivotal role in shaping the direction of this research.

I would also like to acknowledge the Artificial Intelligence And Data Science, VIIT for providing the necessary infrastructure and resources essential for conducting this research. The access to special facilities or equipment significantly contributed to the success of the experimental phase.

Furthermore, I extend my gratitude to my peers and colleagues who have provided valuable input, suggestions, and encouragement at various stages of the project. Their collaborative spirit has enriched the quality of this work.

Last but not least, I am thankful to my family and friends for their unwavering support and understanding during the ups and downs of this project. Their encouragement and motivation have been a constant source of inspiration.

This project would not have been possible without the collective efforts of everyone involved, and for that, I am truly thankful.

Vivek Shinde

Rohan Sonawane

Rohan Jagtap

Suyash Yeolekar

ABSTRACT

The increasing sophistication of audio deep fake technologies presents a serious threat to the integrity of audio communication systems, with potential misuse in identity theft, misinformation, and fraud. The primary aim of this project is to develop a robust detection system capable of distinguishing between genuine and manipulated audio signals to enhance the security of voice based applications. To achieve this, we propose a novel approach that integrates Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory (Bi-LSTM) networks. The CNN component extracts spatial features from audio spectrograms, identifying unique patterns that differentiate real and fake audio. These extracted features are then processed by the Bi-LSTM, which captures temporal dependencies and context, further improving detection accuracy. The proposed model was tested on the “Lj speech dataset” and “wavefake dataset” , demonstrating superior performance over conventional methods with improved generalization across various spoofing techniques. This approach is particularly applicable in enhancing security in voice authentication systems, digital forensics, and other domains requiring reliable detection of audio manipulations.

CONTENTS

LIST OF FIGURES

CHAPTER 01
INTRODUCTION

1.1 Overview :

The project focuses on addressing the growing threat of audio deepfake technology by developing an advanced detection system that can effectively distinguish between authentic and manipulated audio. With applications in areas like voice authentication, digital forensics, and secure communication, the aim is to safeguard against malicious uses of deep fake audio in identity theft, misinformation, and fraud.

1.2 Motivation :

The motivation for this project stems from the rapid advancements and accessibility of audio deepfake technologies, which pose serious risks to the integrity and security of voice based communication. With the increasing ease of creating hyper-realistic synthetic audio, deepfake audio has become a powerful tool that can be exploited in several harmful ways, including identity theft, misinformation campaigns, financial fraud, and the manipulation of public opinion. These risks highlight the urgent need for a reliable detection mechanism to identify fake audio and protect users from malicious misuse.

1.3 Problem Definition and Objectives :

This project aims to address the growing security threat posed by audio deepfake technology, which enables realistic yet fake audio generation that can be misused for identity theft, fraud, and misinformation. The objective is to develop a detection model that combines CNN and Bi-LSTM architectures to distinguish between real and manipulated audio. By processing spatial features with CNNs and capturing temporal patterns with BiLSTM, the model can detect subtle inconsistencies in fake audio, achieving high accuracy and generalization across different spoofing methods. Tested on the LJ Speech and WaveFake datasets, this approach shows promise for enhancing security in voice authentication, digital forensics, and other applications reliant on authentic audio.

1.4 Project Scope & Limitations:

The scope of this project is to develop a robust detection model that combines CNN and Bi-LSTM networks to accurately identify audio deep fakes, targeting applications in voice authentication, digital forensics, and secure communication. Tested on the LJ Speech and WaveFake datasets, the model aims to generalize across various spoofing techniques for practical use in security systems. However, limitations include dependency on diverse datasets, high computational requirements, sensitivity to environmental variability, and challenges in detecting emerging spoofing techniques, all of which may impact real-world applicability and require continuous model updates.

1.5 Methodologies of Problem solving:

- Preprocessing of Audio Data : For effective detection of audio deep fakes, preprocessing is crucial to extract meaningful features from the raw audio. This study employs Mel-Frequency Cepstral Coefficients (MFCCs), a feature extraction method known for its ability to mimic the human ear's response to sound.
- Model Architecture : The proposed model architecture for detecting audio deepfakes combines Convolutional Neural Networks (CNNs) for feature extraction and Bidirectional Long Short-Term Memory (BiLSTM) layers for sequential modeling. This hybrid approach leverages the spatial feature learning capabilities of CNNs and the temporal sequence modeling strength of BiLSTMs
- Model Training : The model is compiled using the Adam optimizer with a learning rate of 0.001, binary cross-entropy loss, and an accuracy metric. The training is conducted with early stopping and learning rate reduction callbacks to optimize convergence

CHAPTER 02
LITERATURE SURVEY

Numerous articles have made a substantial contribution to the development of A Novel Approach to Audio Deep Fake Detection using CNN and Bi-LSTM by utilizing a variety of approaches and strategies to solve the problems in this field. Here, we give a summary of a few important papers in this area.

- "Detecting Audio Deepfakes: Integrating CNN and BiLSTM with Multi-Feature Concatenation" (Taiba Majid Wani, Syed Asif Ahmad Qadri, et al.,2024) [2]: A novel architecture that effectively detects deepfake audio by merging Convolutional Neural Networks (CNN) with Bidirectional Long Short-Term Memory (BiLSTM). To capture spatial and temporal audio properties, our method uses a wide range of acoustic features, including MFCC, Mel spectrograms, CQCC, and CQT, which are processed by CNNs and analyzed by BiLSTM. Our model's excellent accuracy and low Equal Error Rate (EER) have been validated on the ASVSpoof 2019 and FoR datasets, improving the ability to detect audio deepfakes.

- “Audio Deepfake Approaches” (Ousama A. Shaaban; Remzi Yildirim; Abubaker A. Alguttar et al., 2023) [3]: Information regarding generic deep fakes is given in the first half of this study, which reviews methods used in the production and detection of audio deepfakes. In the second part, the primary techniques for audio deepfakes are described and then contrasted. The findings cover a number of techniques for identifying audio deepfakes, such as applying machine learning and deep learning algorithms, researching media consistency, and assessing statistical characteristics. In these investigations, the main techniques for identifying phony audio were Support Vector Machines (SVMs), Decision Trees (DTs), Convolutional Neural Networks (CNNs), Siamese CNNs, Deep Neural Networks (DNNs), and a CNN-Recurrent Neural Networks (RNNs) combination. These techniques differed in their accuracy; SVM had the best accuracy at 99%, while DT had the lowest at 73.33%.

- “Deepfake Audio Detection Using Spectrogram-based Feature and Ensemble of Deep Learning Models” (Lam Pham; Phat Lam et al., 2024) [4]: We suggest a deep learning-based method for identifying deepfake audio that was created as a component of the EUCINF (EUropean Cyber and INFormation) project, a multi-partner European endeavor. Our method uses Wavelet Transform (WT), Constant-Q Transform (CQT), and Short-Time Fourier Transform (STFT) with several auditory filters (Mel, Gammatone, linear, DCT) to convert raw audio into spectrograms. A Multilayer Perceptron (MLP) is used to analyze audio embeddings from pre-trained models (Whisper, Seamless, Speech Brain, Pyannote), (2) transfer learning with vision models (e.g., ResNet-18, MobileNet-V3, EfficientNet), and (3) baseline CNN, RNN, and C-RNN models. The efficiency of spectrogram modifications and deep learning in audio deepfake detection is demonstrated by the Equal Error Rate (EER) of 0.03 obtained from the fusion of top-performing models on the ASVspoof 2019 dataset.

- "Audio Replay Spoof Attack Detection by Joint Segment-Based Linear Filter Bank Feature Extraction and Attention-Enhanced DenseNet-BiLSTM Network" (Lian Huang; Chi-Man Pun et al.,2020) [5]: Automatic speaker verification (ASV) systems are not very good at spotting replay attempts and are quite vulnerable to spoofing attacks. We suggest an attention-enhanced DenseNet-BiLSTM model that uses segment-based linear filter bank features to solve this. Using short-term zero-crossing rate and energy, our method first finds the speech signal's quiet segments. If there is not enough silent data, decaying tails are chosen. These segments are then used to extract characteristics of a high-frequency linear filter bank. The BTAS2016 and ASVspoof2017 datasets were used to test the DenseNet-BiLSTM architecture, which has been improved with attention techniques to reduce overfitting. Relative performance increases of 91.68% and 74.04% on the corresponding datasets demonstrate a notable improvement over baseline systems.
- "Detecting Fake Audio of Arabic Speakers Using Self-Supervised Deep Learning" (Zaynab M. Almutairi; Hebah Elgibreen et al.,2023) [6]: Audio deepfakes, where AI is used to clone voices, pose significant threats to public safety. While Machine Learning (ML) and Deep Learning (DL) techniques exist for detection, they often require large datasets and extensive pre-processing. To address these issues, we propose Arabic-AD, a self-supervised learning based method for detecting synthetic and imitated Arabic speech. This work introduces the first synthetic dataset of a single speaker fluent in Modern Standard Arabic (MSA) and includes Arabic recordings from non-native speakers to assess robustness. In experiments, Arabic-AD achieved superior performance with a 0.027% Equal Error Rate (EER) and 97% accuracy, outperforming existing benchmarks without requiring excessive data.
- "A Hybrid CNN-BiLSTM Voice Activity Detector" (Nicholas Wilkinson; Thomas Niesler et al., 2021) [7]: This paper introduces a hybrid architecture for voice activity detection (VAD) combining convolutional neural networks (CNN) and bidirectional long short-term memory (BiLSTM) layers, trained end-to-end for improved efficiency in noisy, under-resourced environments. We use nested k-fold cross-validation to explore hyperparameters and balance model size with performance. Our system, tested on the AVA-Speech dataset, outperforms three baselines, including a larger ResNet model, with an AUC of 0.951, especially in challenging noise conditions. BiLSTM layers provide a ≈2% accuracy improvement over unidirectional LSTM layers.
- "Audio Deepfakes: Feature Extraction and Model Evaluation for Detection" (Ramesh K Bhukya; Aditya Raj; Danish N Raja et al.,2024) [8]: the detection of audio deepfakes, which pose significant risks to public safety. Using feature extraction techniques like Mel-frequency cepstral coefficients, chromagrams, and Mel-spectrograms, we train machine learning and deep learning models on the ASVspoof2021 dataset to classify speech as bonafide or spoofed. Experimental results show the multilayer perceptron achieves 96.07% accuracy, while SVM, k-NN, XGBoost, and RF yield accuracies between 93.54% and 94.58%. Deep learning models,

particularly CNN, outperform others with an accuracy of 98.2%, highlighting its effectiveness in detecting deepfake audio.

- "A Survey on the Detection and Impacts of Deepfakes in Visual, Audio, and Textual Formats" Rami Mubarak; Tariq Alsboui et al.,2023) [9]: This study explores the growing threat of deepfakes—fake visual, audio, and textual content—created using AI techniques. While much focus has been on visual and audio deepfakes, text-based deepfakes are also becoming a significant concern due to advancements in natural language processing. The study reviews the social, political, economic, and technological impacts of deepfakes and critically evaluates current detection methods. It calls for unified, real-time solutions and emphasizes the need for a holistic approach combining technical measures, public awareness, and legislative action to address the challenges of deepfakes.
- "Audio-deepfake detection: Adversarial attacks and countermeasures" (Mouna Rabhi, Spiridon Bakiras , Roberto Di Pietro et al., 2024) [10]: This study examines the vulnerability of AI-based audio authentication systems to adversarial attacks using deepfake audio. We demonstrate that state-of-the-art audio deepfake classifiers, such as the Deep4SNet model with 98.5% accuracy, can be severely compromised. Our two adversarial attacks, leveraging generative adversarial networks, reduce the detector's accuracy to nearly 0%, specifically dropping it to 0.08% in a graybox scenario. To counter these attacks, we propose a lightweight, generalizable defense mechanism adaptable to any audio deepfake detector and outline future research directions for enhancing security in audio-based authentication systems.

This project builds on the current research by proposing a novel hybrid CNN-Bi-LSTM architecture optimized for real-world noise conditions, with a focus on efficiency and resilience to adversarial attacks. Through a combination of CNN and Bi-LSTM layers, the approach seeks to capture both spatial and sequential dependencies, aiming to improve detection rates in noisy environments and against adversarial manipulation

CHAPTER 03
SOFTWARE REQUIREMENTS SPECIFICATION

3.1 Introduction

The Audio Deepfake Detection System is designed to identify and detect deepfake audio generated by AI models. This system is built using advanced machine learning algorithms, primarily leveraging Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory (Bi-LSTM) networks. The project focuses on processing audio signals to identify alterations that may indicate deep fake content. The system accepts audio samples, extracts features like Mel-frequency Cepstral Coefficients (MFCC), and applies machine learning models to classify whether the audio is genuine or artificially manipulated.

The system is developed with simplicity and efficiency in mind. It features a Streamlit based frontend, where users can upload audio files for analysis. The core processing is carried out using Python-based tools and machine learning frameworks, without the need for complex backend systems, databases, or external integrations. The system does not currently require cloud storage, APIs, or user authentication features such as login or password management. It is designed to be a standalone solution that can be easily used for one-time analysis of audio files.

3.2 User Classes and Characteristics

The Audio Deepfake Detection System is designed with a specific user class in mind:

- Description: The primary users of the system are professionals, researchers, or individuals who need to verify the authenticity of audio samples. They may be journalists, security personnel, or anyone who works with media content and needs to detect deepfake manipulations.
- Characteristics:
  - Comfortable using digital tools for uploading and interacting with files.
  - Expect a fast and efficient system for analyzing audio files.
  - Require a user-friendly interface with simple instructions.
  - May have varying technical expertise but expect a tool that delivers results with minimal technical input.

3.3 Operating Environment

The system will operate in the following environment:

- Frontend: The frontend will be built using Streamlit, a Python framework designed for quick web applications. This allows users to upload and interact with audio files through a simple interface. The frontend will display results, including whether the audio file is classified as genuine or deepfake.

- Backend: The backend will be powered by Python, leveraging machine learning libraries such as TensorFlow or PyTorch for training and deploying the audio detection model. Feature extraction will be done using librosa for MFCC extraction and other audio processing tasks.
- Storage: No external database or cloud storage is required. The system will handle audio file uploads directly and process them without storing files in a database.
- Model: The core model will use CNN and Bi-LSTM for detecting deepfakes in audio files. The model will be trained on datasets containing both real and manipulated audio data to distinguish between the two.

3.4 External Interface Requirements

1. User Interfaces (UI)

- Frontend Interface:
  - Platform: The application will be accessible through a web browser.
  - Design Principles:
    - Simple Design: The interface will be minimalist with a drag-and-drop or file picker option for uploading audio files.
    - Results Display: After processing, the system will display the classification result (Real or Fake) with a confidence score.
    - Real-Time Feedback: Users will see the status of the analysis process (e.g., "Processing..." or "Analysis Complete").
    - Result Download: The application will allow users to download the analysis results, possibly in a text format, displaying the classification details and confidence scores.

2. Hardware Requirements:

- Devices: Desktop PCs or laptops capable of running modern web browsers.
- Browser: Google Chrome, Mozilla Firefox, or Microsoft Edge.

3. Software Tools:

- Frontend Development:
  - Streamlit: For building the interactive web-based interface.
- Backend Development:
  - Python: Main programming language used for backend processing.
  - TensorFlow/PyTorch: For machine learning model deployment.
  - librosa: For audio feature extraction, especially MFCC.
  - NumPy, SciPy: For numerical and signal processing tasks.
- Audio Processing:
  - librosa: For extracting audio features like MFCC from raw audio files.

3.5 Functional Requirements

- Audio File Upload:
  - The system shall allow users to upload audio files in commonly used formats (e.g., WAV, MP3).
  - The system shall process the uploaded file and extract the relevant audio features (e.g., MFCC).

- Dataset Requirements:
  - The system shall use high-quality datasets like LJ Speech and WaveFake Dataset with real and deep fake audio.
  - The system shall extract MFCC features and apply data augmentation techniques to enhance model robustness.

- Deep Face Detection:
  - The system shall use a trained deep learning model (CNN + Bi-LSTM) to classify the uploaded audio as either genuine or deepfake.
  - The system shall output a confidence score along with the classification result.

- Results Display:
  - The system shall display the classification result (Real or Fake) to the user.
  - The system shall display a confidence score based on the model's certainty about the classification.

- Result Download:
  - Users shall be able to download the classification result, which includes the label (Real or Fake) and the confidence score.

3.6 Non-Functional Requirements

1. Performance Requirements:

- The system should process audio files and provide results within a reasonable time, ideally under 10 seconds per file for typical audio durations (e.g., 1-5 minutes).

2. Scalability:

- The system is not designed for heavy traffic or scalability concerns as it is intended for small, one-time use cases with a focus on accuracy rather than load handling.

3. Security:

- The system does not require advanced security features such as encryption, authentication, or role-based access control.
- Since it is a simple standalone application, security considerations are minimal.

4. Usability:

- The system should be easy to use with a clear, user-friendly interface that requires no specialized knowledge from the user.
- The application will not require users to have any background in audio processing or machine learning.

5. Maintainability:

- The system's codebase should be simple, well-commented, and modular for easy maintenance and future improvements.
- Future versions may incorporate more advanced features, such as batch processing or cloud-based deployment.

6. Compatibility:

- The application should be compatible with modern web browsers and responsive to different screen sizes (desktop, tablet, and mobile).

3.7 Product Perspective

The Audio Deepfake Detection System is a standalone tool designed for one-time, small-scale analysis of audio files. It is a simple solution for detecting deep fake audio using deep learning models and does not require a database, cloud storage, or third-party API integrations. The system will work efficiently on local systems with Streamlit as the user interface and Python-based machine learning models for detection.

3.8 Product Function

The product is focused on analyzing audio files and detecting potential deepfake manipulation. The system allows users to upload audio files, from which the system extracts relevant features and processes them using a pre-trained deep face detection model. The results, including classification and confidence scores, are displayed in the user interface, allowing users to easily identify whether the audio is genuine or deepfake.

CHAPTER 04
SYSTEM DESIGN

4.1 System Architecture :

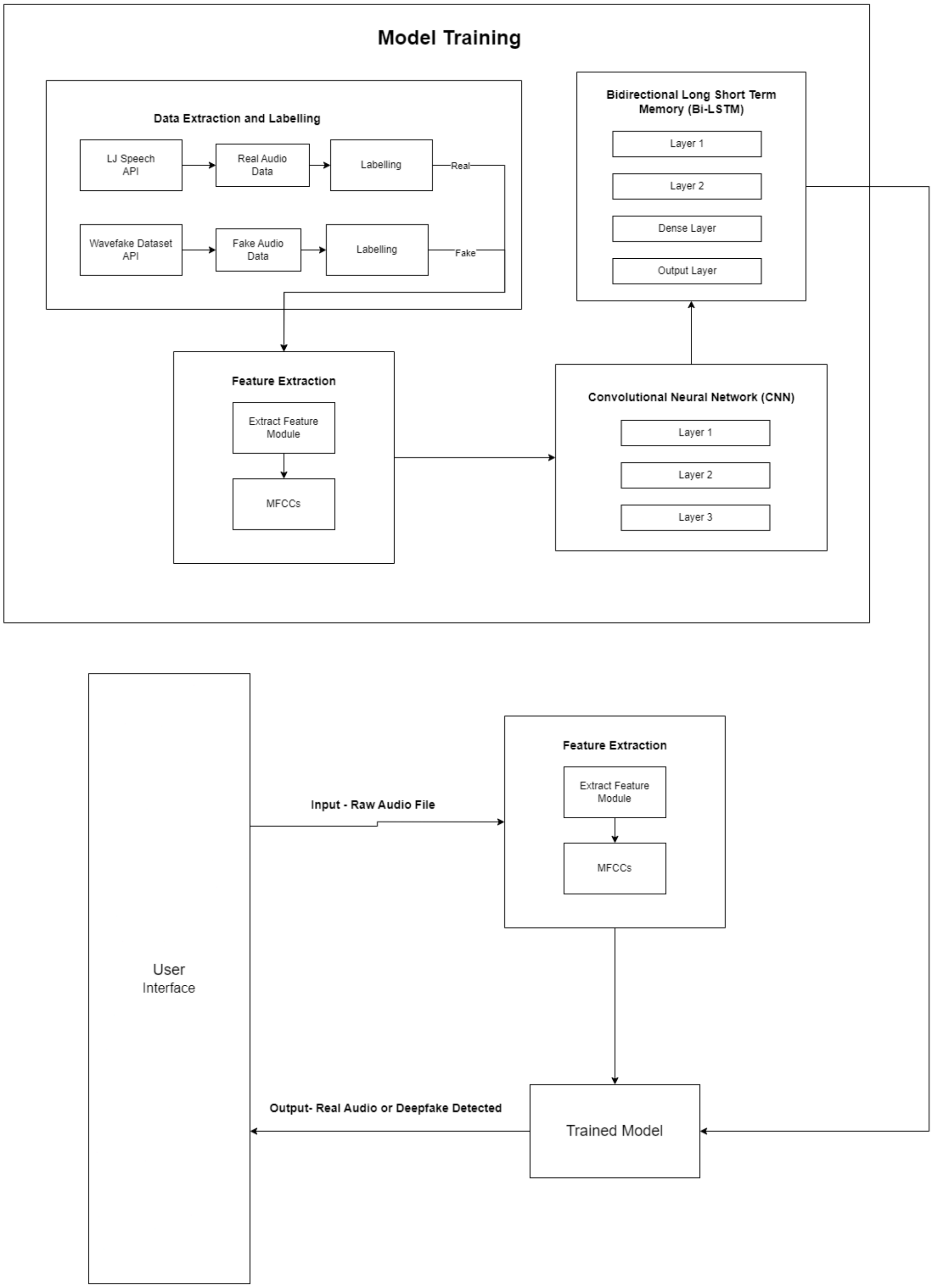


Fig.4.1 System Architecture

4.2 Data Flow Diagram:

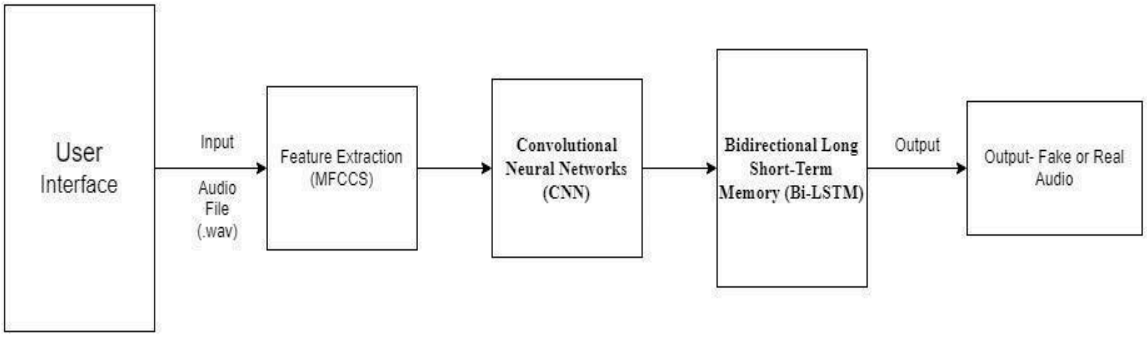

User Interface
Input
Audio File (.wav)
Feature Extraction (MFCCS)
Convolutional Neural Networks (CNN)
Bidirectional Long Short-Term Memory (Bi-LSTM)
Output
Output- Fake or Real Audio

Fig.4.2 Data Flow Diagram

4.3 Use Case Diagram:

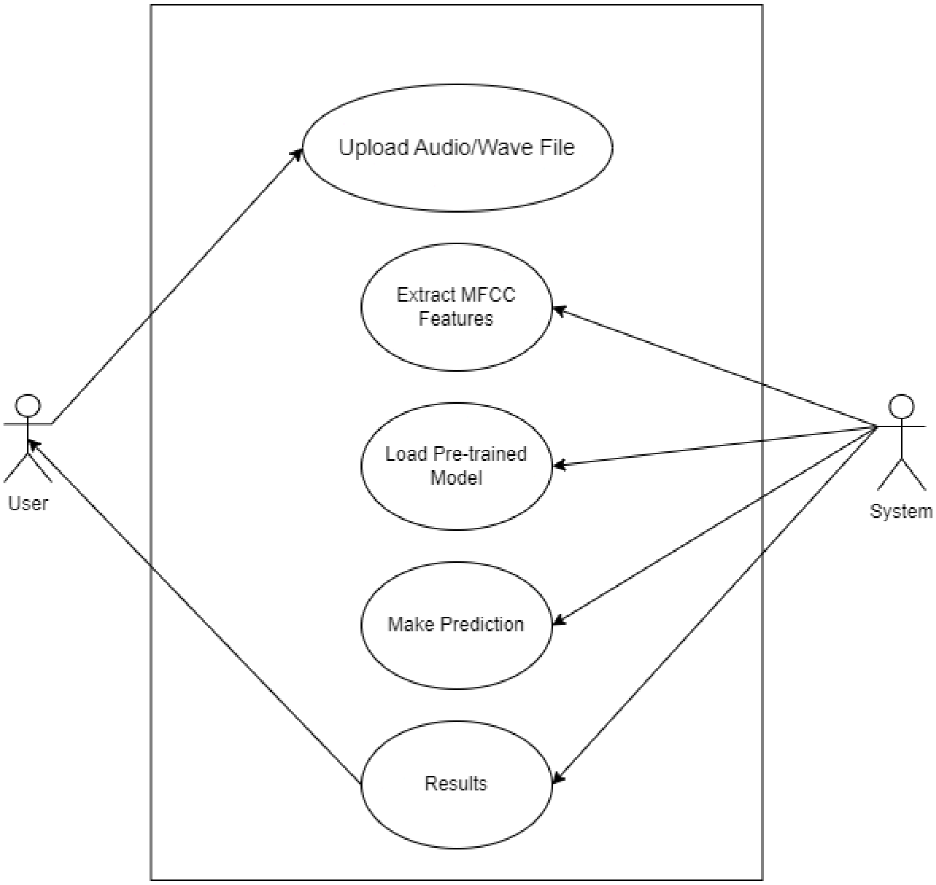
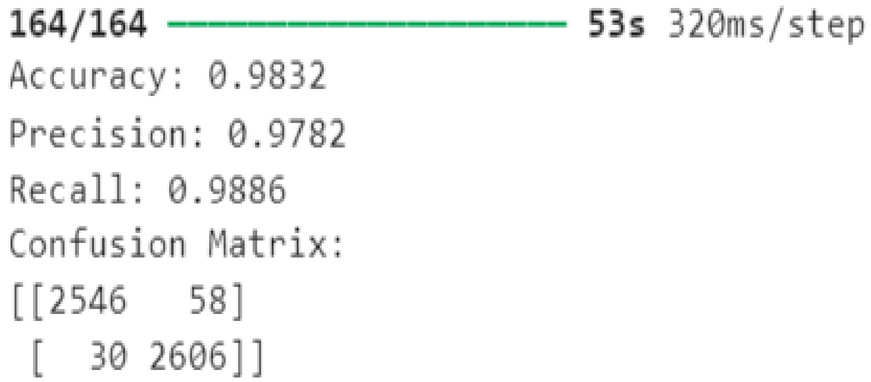


Fig.4.3 Use Case Diagram

4.4 Model Training and Evaluation:



```
164/164 ━━━━━━━━━━━━━━━━━━━━ 53s 320ms/step
Accuracy: 0.9832
Precision: 0.9782
Recall: 0.9886
Confusion Matrix:
[[2546   58]
 [  30 2606]]
```

Fig.4.4.1 Precision, Recall and Accuracy

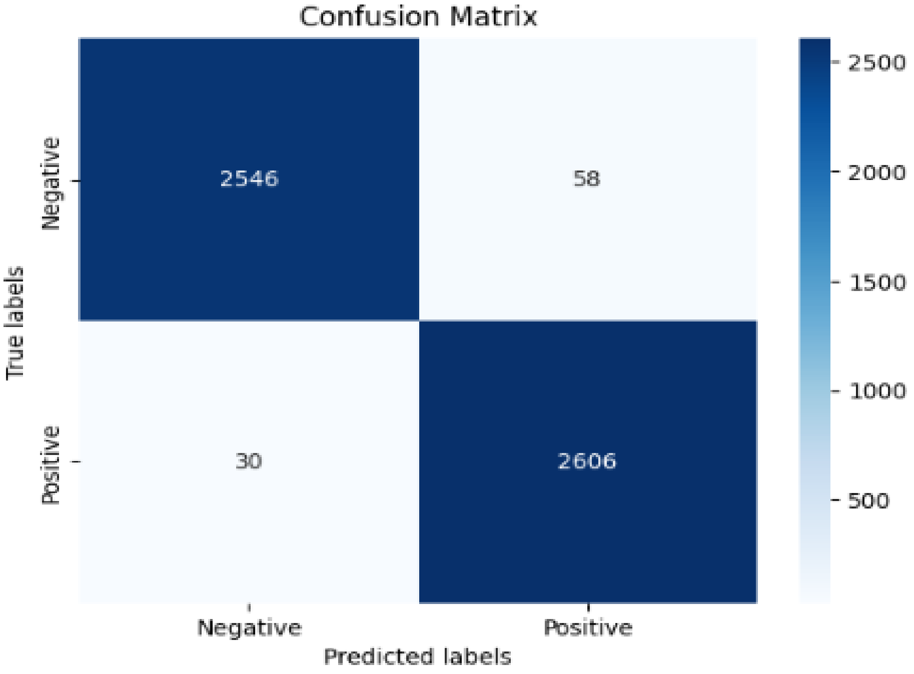


Fig.4.4.2 Confusion Matrix

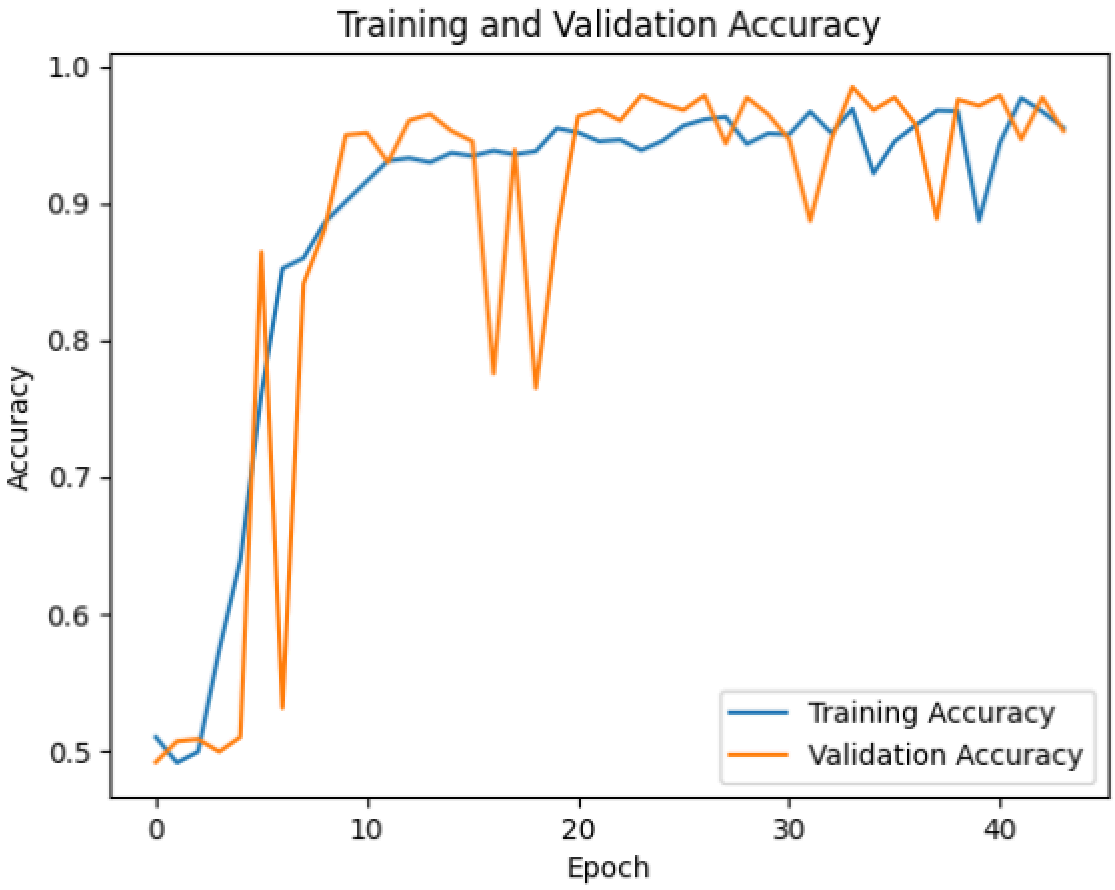


Fig.4.4.3 Training Accuracy vs Validation Accuracy Graph

CHAPTER 05
PROJECT PLAN

5.1 Project Estimate:

The project estimate for Audio Deep Face Detection focuses on the costs related to resources required for training the models and deploying the system. This includes data acquisition, computing resources, personnel, and contingency provisions:

i. Data Collection and Preprocessing:

- Costs for acquiring datasets such as WaveFake and LJ Speech Dataset.
- Preprocessing tools for feature extraction (e.g., MFCC) and data augmentation.

ii. Computing Resources:

- Infrastructure costs for model training, including GPU/TPU access or cloud services.

iii. Contingency Budget:

- Provision for unexpected requirements like additional data or extended computational resources.

5.2 Risk Management:

i. Technical Risks:

- Risk: Challenges in training deep learning models for deep fake audio detection.
- Mitigation: Conduct extensive prototyping and model validation. Use pre-trained models for faster convergence and evaluate different architectures.

ii. Resource Risks:

- Risk: Insufficient access to high-quality labeled datasets or computational power.
- Mitigation: Leverage publicly available datasets and cloud services. Seek external expertise for specific technical issues.

iii. Timeline Risks:

- Risk: Delays in model training or integration of deepfake detection components.
- Mitigation: Break down the project into manageable tasks with regular reviews. Use agile methodologies to adjust the schedule as needed.

iv. Usability and Output Risks:

- Risk: False positives or incorrect classification of audio as deepfake.
- Mitigation: Improve the training dataset, conduct rigorous testing, and optimize the model for high accuracy.

5.3 Project Schedule:

1. **Planning Phase (Week 1):**
   - Define project objectives, deliverables, and scope.
   - Review literature on audio deep face detection techniques.
   - Develop a detailed project timeline and task breakdown.
2. **Data Acquisition and Preprocessing (Week 2-3):**
   - Acquire datasets (LJ Speech, WaveFake Dataset) and preprocess them.
   - Extract audio features (MFCC) and apply data augmentation techniques.
3. **Model Implementation and Training (Week 4-5):**
   - Implement deep learning models (CNNs, Bi-LSTM) for deepfake detection.
   - Train and validate models using the preprocessed data.
   - Conduct hyperparameter tuning to improve model performance.
4. **Integration and Testing (Week 6):**
   - Integrate the trained model into a unified detection system.
   - Test the system with real-world audio samples and evaluate its accuracy.
5. **Evaluation and Optimization (Week 7):**
   - Evaluate model performance using key metrics (e.g., accuracy, F1-score, precision).
   - Optimize the model for speed and efficiency in real-time detection.
   - Address any shortcomings identified in testing.
6. **Report Writing and Documentation (Week 8):**
   - Document the methodology, results, and conclusions.
   - Prepare a final project report and user guide for deploying the detection system.
7. **Contingency (Week 9):**
   - Additional time for addressing unforeseen challenges, such as retraining models or troubleshooting integration issues.

5.4 Team Organization:

1. **Data Analyst:**
   - Collect and preprocess datasets.
   - Conduct feature extraction (e.g., MFCC) and data augmentation.
   - Perform data analysis to support model development.
2. **Machine Learning Engineer:**
   - Implement and train the deep fake detection model (CNNs, Bi-LSTM).
   - Conduct hyperparameter tuning and evaluate model performance.
   - Provide detailed experimental results for documentation.
3. **Audio Processing Expert:**
   - Optimize audio feature extraction techniques for deep face detection.
   - Collaborate with the team to improve model robustness and accuracy.
4. **Software Developer:**
   - Develop the user interface for the deepfake detection system.
   - Integrate the trained model with the application backend.
   - Test and optimize system performance for real-time audio detection.
5. **System Integrator:**
   - Ensure the integration of all system components, including the frontend and model.
   - Conduct end-to-end testing and ensure real-time performance.
   - Manage deployment and maintain system functionality.

CHAPTER 06
PROJECT IMPLEMENTATION

6.1. Overview of Project Modules

The project for detecting audio deepfakes is divided into several structured modules, ensuring clarity and modularity during implementation. Each module has specific responsibilities and integrates seamlessly to achieve the final output.

1. **Data Preprocessing Module**:
   - ○ This module is responsible for handling the uploaded audio files and preparing them for the model.
   - ○ Tasks include:
     - ■ **Audio Loading**: The uploaded file is read and converted into an array using Librosa.
     - ■ **Feature Extraction**: Mel-frequency cepstral coefficients (MFCCs) are computed from the audio data to capture key audio features.
     - ■ **Normalization and Reshaping**: The MFCC features are padded or truncated to a fixed size of (40, 500) to ensure consistency for the deep learning model input.

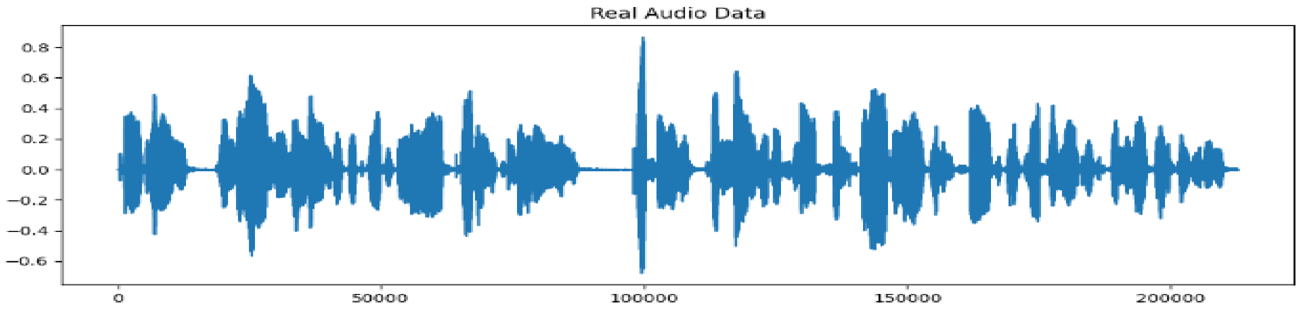


Fig.6.1.1 Real Audio Frequency domain grapH

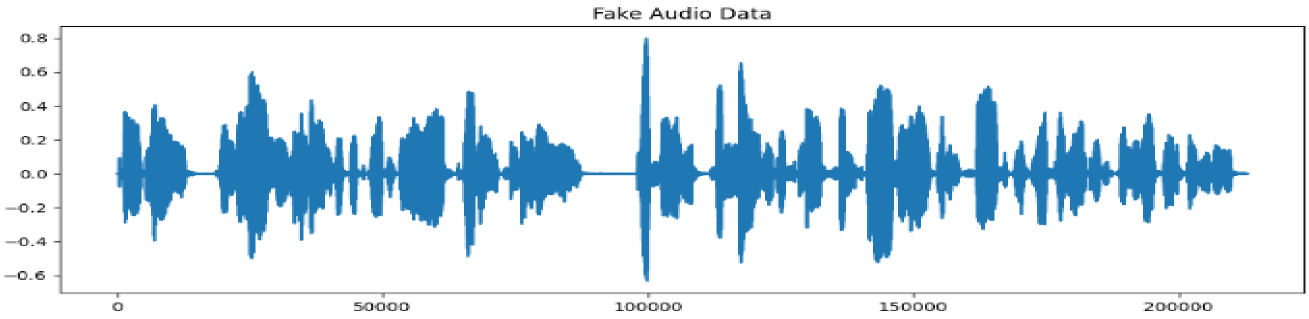


Fig.6.1.2. Fake Audio Frequency domain graph

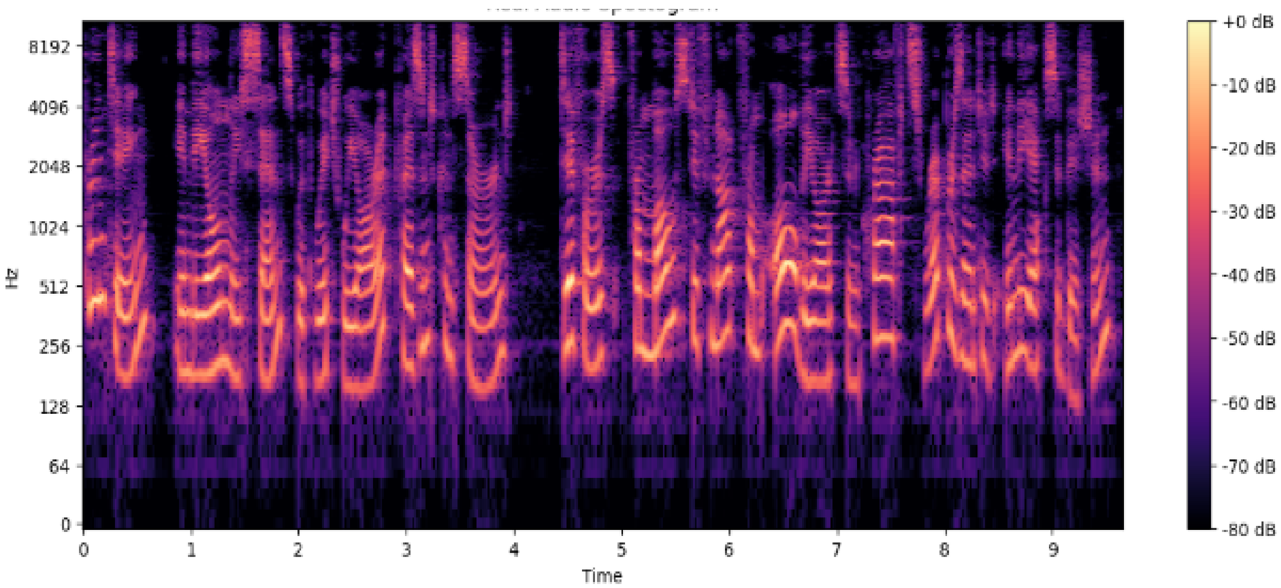


Fig.6.1.3. Real Audio Spectrogram

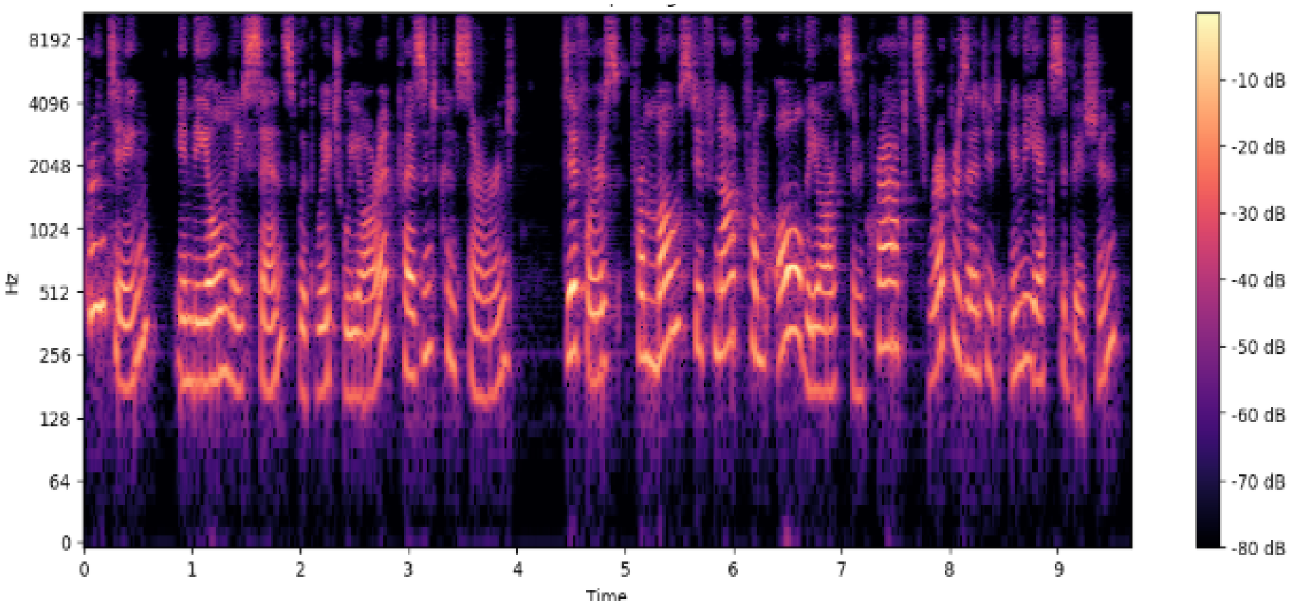


Fig.6.1.4. Fake Audio Spectrogram

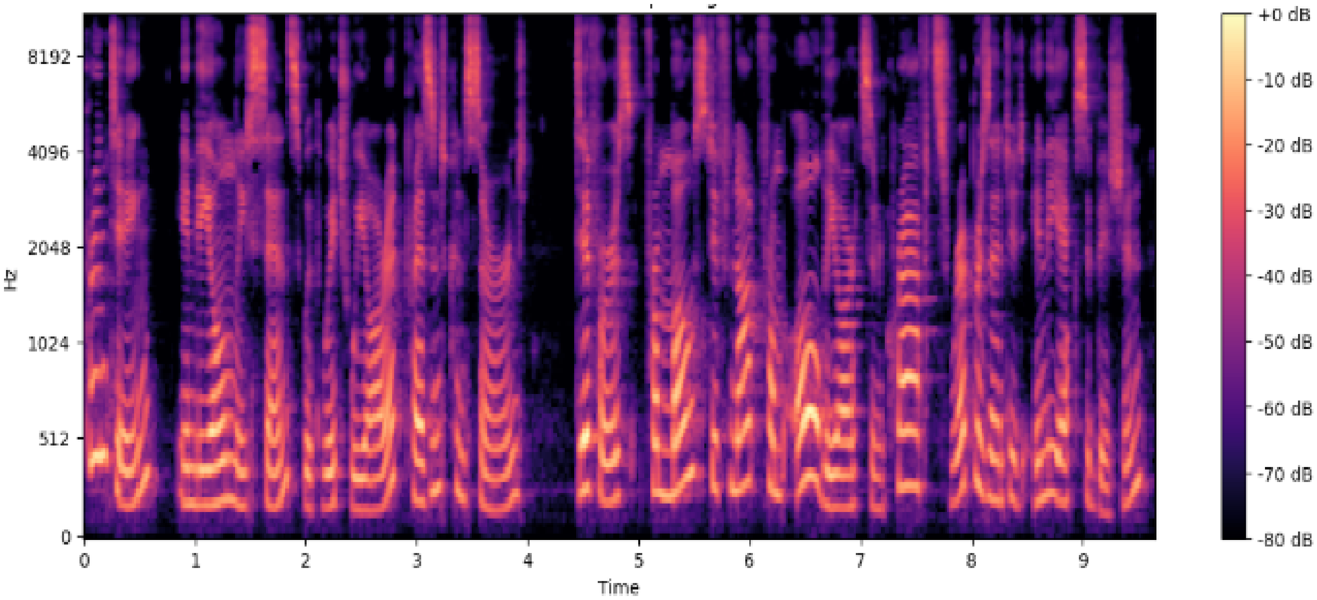


Fig.6.1.5. Real Audio Mel Spectrogram

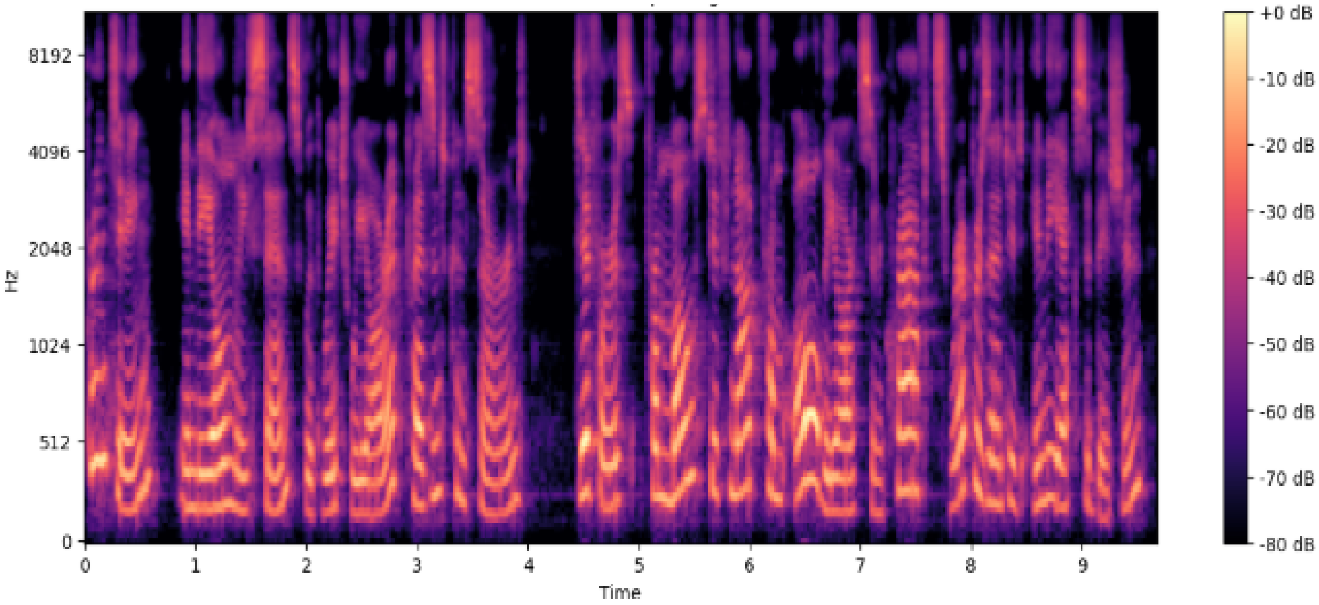


Fig.6.1.6. Fake Audio Mel Spectrogram

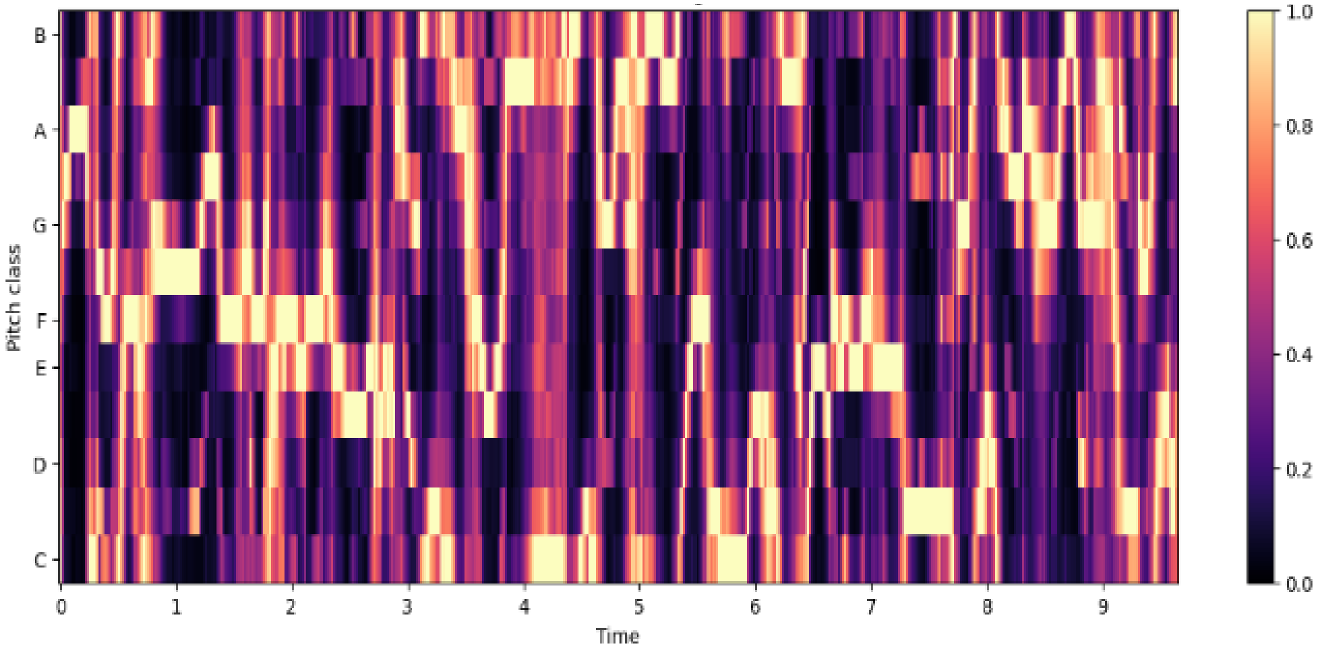


Fig.6.1.7. Real Audio Chromagram

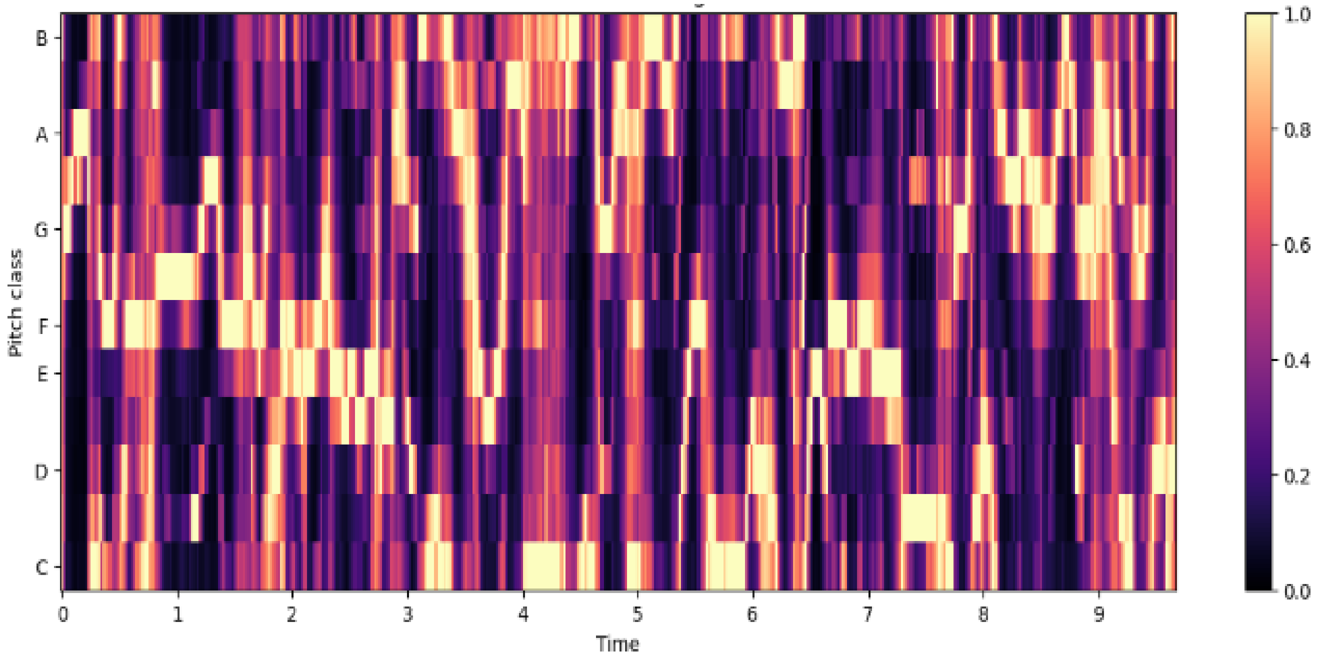


Fig.6.1.8. Fake Audio Chromagram

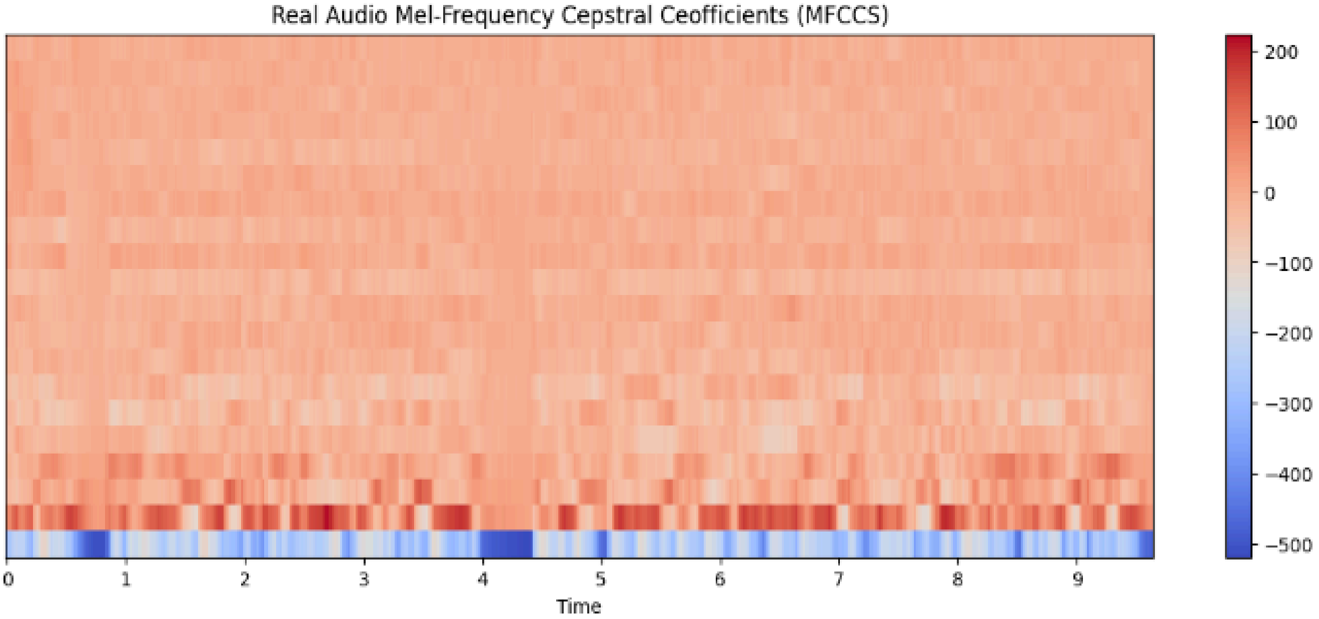


Fig.6.1.9. Real Audio MFCCs

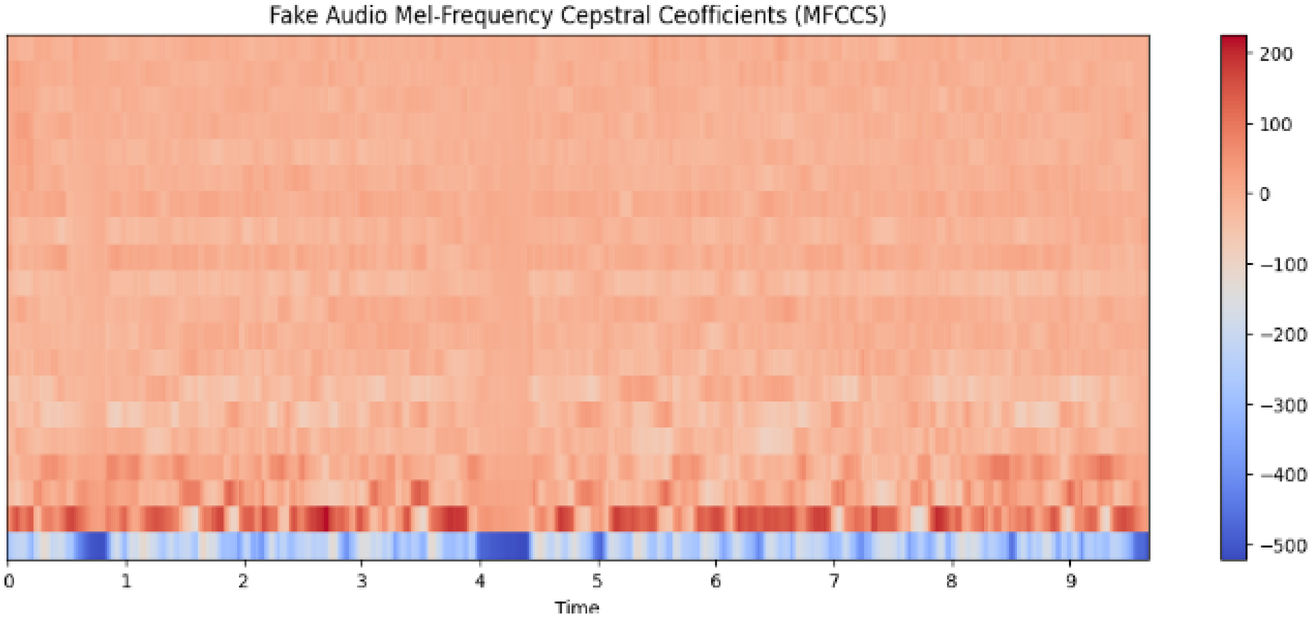


Fig 6.1.10. Fake Audio MFCCs

2. **Model Training Module**:
    - This module defines, trains, and evaluates the deep learning model.
    - Steps involved:
        - **Model Architecture**: Combines Convolutional Neural Networks (CNNs) for feature extraction and Bi-Long Short-Term Memory (Bi-LSTM) layers for learning temporal patterns in the data.
        - **Training Process**: The model is trained on labeled real and synthetic (deepfake) audio datasets, ensuring it learns to differentiate between the two.
        - **Validation and Tuning**: Performance is measured using metrics like precision, recall, and F1-score. Hyperparameters such as learning rate, batch size, and number of epochs are optimized.
3. **Prediction Module**:
    - After the model is trained, this module is responsible for using the model to generate predictions.
    - Tasks include:
        - Processing user-uploaded audio files.
        - Extracting MFCC features similar to the preprocessing stage.
        - Feeding the processed features into the trained model to get a classification output (real or deep fake) and the confidence score.
4. **Web Application Module**:
    - Built using Streamlit, this module provides a user-friendly graphical interface for users to interact with the system.
    - Features include:
        - Uploading an audio file in common formats like .wav, .mp3, and .ogg.
        - Displaying an audio player for users to listen to the uploaded file.
        - Showing the prediction results with a confidence percentage and visualizing the confidence level using a progress bar.
        - Providing clear error messages and guidance to users when issues occur.

6.2. Tools and Technologies Used

- **Programming Language**: Python was chosen for its extensive libraries and community support in AI/ML.
- **Deep Learning Framework**: TensorFlow/Keras was used to design, train, and deploy the neural network model.
- **Feature Extraction**: Librosa was utilized to compute MFCCs and handle audio data preprocessing.
- **Web Development**: Streamlit provided a lightweight framework to create an interactive, web-based user interface for the application.

- **Libraries for Data Handling**: NumPy and Pandas were used for managing arrays and tabular data structures.
- **Model Deployment**: The model was deployed and tested in a local environment, ensuring smooth integration with the Streamlit interface.
- **IDE**: VS Code was the primary development environment, offering debugging, version control, and plugin support.

6.3. Algorithm Details

The heart of the project lies in its carefully designed algorithm, which integrates advanced signal processing and deep learning techniques.

1. **MFCC Feature Extraction**:
   - Why MFCCs?
     MFCCs are a compact representation of the audio signal, mimicking how humans perceive sound. They capture important frequency characteristics that are critical for differentiating real and synthetic audio.
   - **Steps in Feature Extraction**:
     - Audio signals are divided into short frames.
     - Each frame undergoes a Fourier Transform to compute a spectrogram.
     - The spectrogram is then mapped onto the Mel scale, emphasizing frequencies most relevant to human hearing.
     - Logarithms are applied to compress the dynamic range, followed by a Discrete Cosine Transform (DCT) to compute MFCCs.
   - The extracted MFCCs are reshaped into (40, 500) arrays, representing the audio's time-frequency features.
2. **Model Architecture**:
   The model combines the strengths of CNNs and LSTMs to effectively handle both spatial and temporal patterns in audio data:
   - **CNN Layers**:
     - Extract spatial features from the MFCCs, identifying local patterns in the time-frequency representation.
     - Includes Conv2D layers followed by MaxPooling2D layers for dimensionality reduction and feature selection.
   - **LSTM Layers**:
     - Capture sequential dependencies and temporal patterns across the extracted features.
   - **Dense Layers**:
     - Fully connected layers aggregate features learned by CNN and LSTM layers, leading to final classification.

- **Output Layer**:
    - A single neuron with a sigmoid activation function outputs a probability score between 0 and 1, indicating the likelihood of the audio being a deep fake.

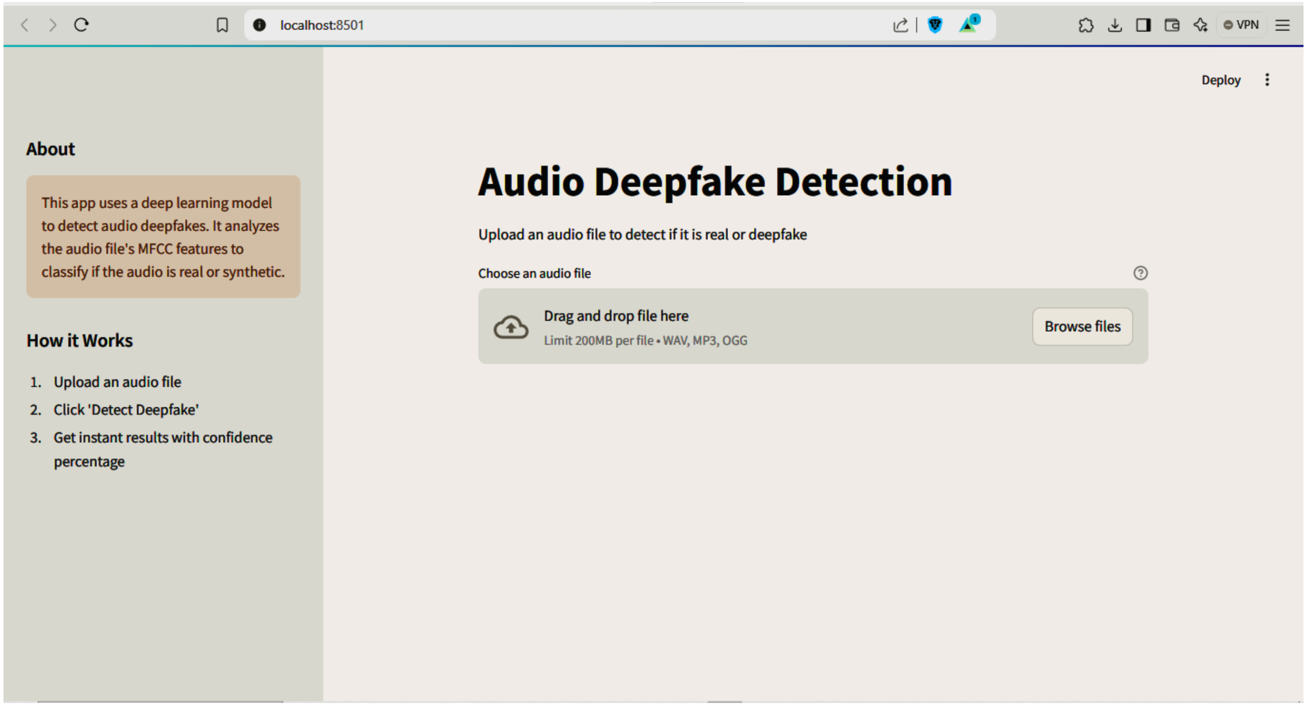
3. **Training Details**:
    - **Loss Function**: Binary Cross-Entropy is used to optimize the classification task.
    - **Optimizer**: The Adam optimizer is selected for its efficient convergence.
    - **Metrics**: Accuracy, precision and recall are calculated to evaluate model performance.
    - **Data Augmentation**: Techniques like adding noise, pitch shifting, or time stretching are applied to the training data to improve generalization.
4. **Prediction Process**:
    - The user-uploaded audio is processed through the same MFCC feature extraction pipeline.
    - The extracted features are input into the trained model, which outputs:
        - A confidence score (0 to 1).
        - A classification:
            - Real Audio if the confidence score is below 0.5.
            - Deepfake Audio if the confidence score is 0.5 or higher.

CHAPTER 07
RESULT

7.1 Outcome

The outcomes of this experiment provide a convincing illustration for upcoming applications that seek to combat deep fake audio in the areas of media integrity and security. Furthermore, by demonstrating the efficacy of customized deep learning techniques, our study adds to the larger conversation on thwarting synthetic media. In order to improve the model's generalizability, future iterations could benefit from investigating more audio features, using a variety of datasets, and implementing ensemble strategies that combine this method with other detection techniques. Finally, our work lays the groundwork for future advancements in this crucial field by highlighting the potential of sophisticated machine learning in preserving authenticity and reliability in audio transmissions.

7.2 Screenshots



About

This app uses a deep learning model to detect audio deepfakes. It analyzes the audio file's MFCC features to classify if the audio is real or synthetic.

How it Works

1. Upload an audio file
2. Click 'Detect Deepfake'
3. Get instant results with confidence percentage

Audio Deepfake Detection

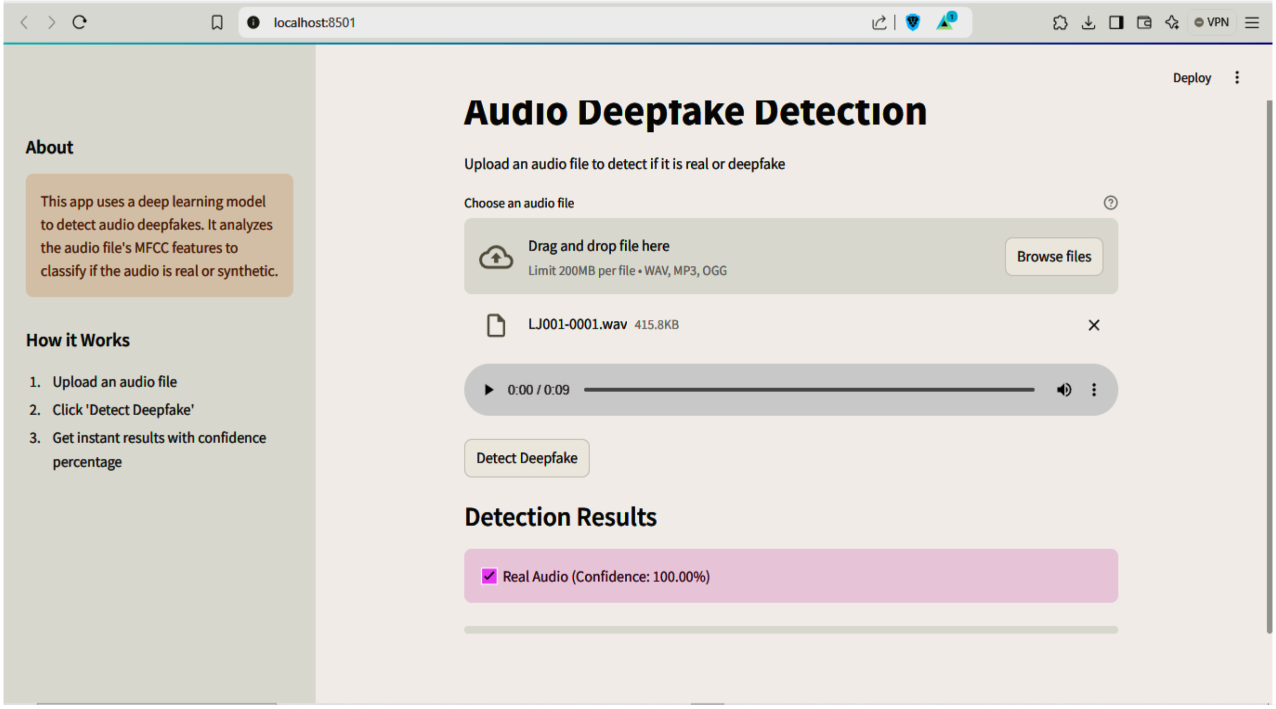
Upload an audio file to detect if it is real or deepfake

Choose an audio file

Drag and drop file here

Limit 200MB per file • WAV, MP3, OGG

Fig.7.2.1 User Interface



Audio Deepfake Detection

Upload an audio file to detect if it is real or deepfake

About

This app uses a deep learning model to detect audio deepfakes. It analyzes the audio file's MFCC features to classify if the audio is real or synthetic.

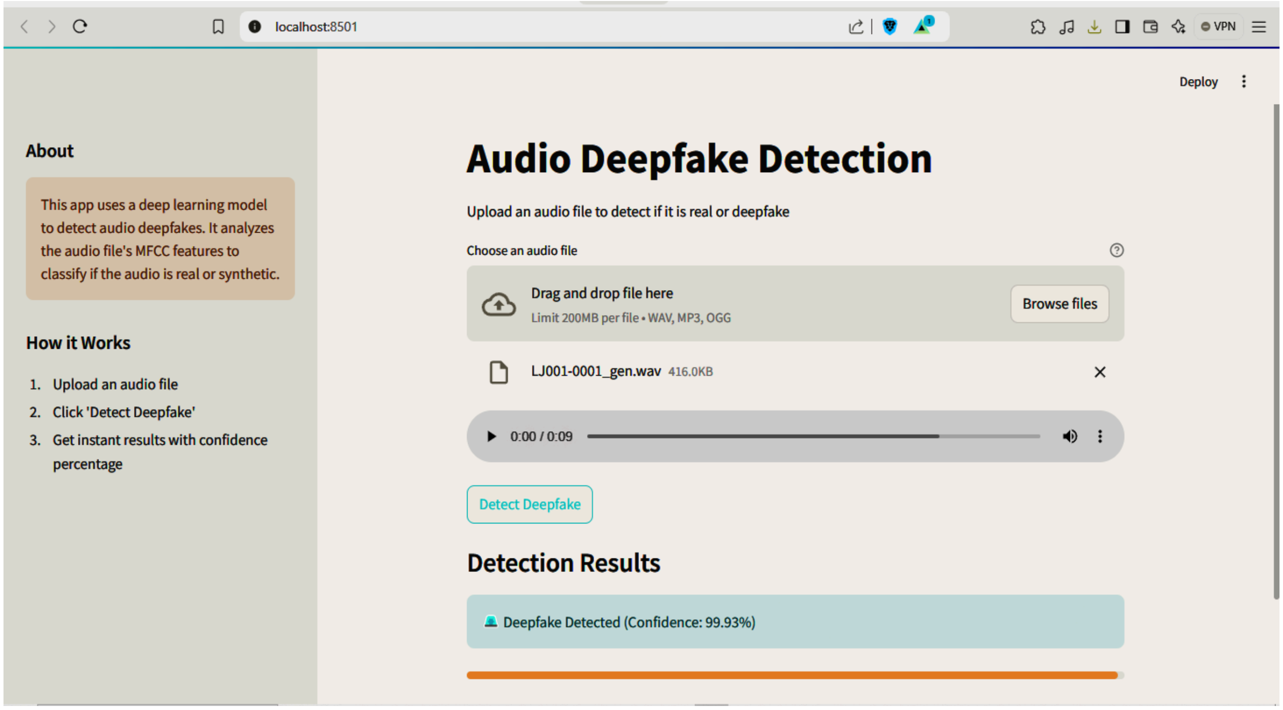
How it Works

1. Upload an audio file
2. Click 'Detect Deepfake'
3. Get instant results with confidence percentage

Detection Results

☑ Real Audio (Confidence: 100.00%)

Fig.7.2.2 Prediction on real Audio



Audio Deepfake Detection

Upload an audio file to detect if it is real or deepfake

About

This app uses a deep learning model to detect audio deepfakes. It analyzes the audio file's MFCC features to classify if the audio is real or synthetic.

How it Works

1. Upload an audio file
2. Click 'Detect Deepfake'
3. Get instant results with confidence percentage

Detection Results

Deepfake Detected (Confidence: 99.93%)

Fig.7.2.3 Prediction on Deep Fake Audio

CHAPTER 08
CONCLUSION

8.1 Conclusion

In this study, we propose a novel deep learning-based method for deep fake audio detection, addressing the crucial problem of differentiating between real and altered audio. Our approach combines Convolutional Neural Networks (CNN) with Bidirectional Long Short-Term Memory (Bi-LSTM) networks in a unique way to leverage both spatial and temporal features in audio data. This builds upon traditional detection methods, which frequently rely on basic signal processing or simpler machine learning algorithms. We close a significant gap in detection techniques by using Mel-frequency Cepstral Coefficients (MFCC) as input characteristics, producing an integrated strategy that improves detection accuracy and dependability. Our results demonstrate the durability of our model in real-world applications by showing notable gains in detecting minute audio irregularities that distinguish real speech from artificial manipulations. Considering the system's notable accuracy and versatility, there are still certain drawbacks, including the requirement for ongoing data augmentation to accommodate developing deepfake methods and issues with the scalability of real-time processing. Despite these difficulties, our work opens the door for more research avenues, such as the investigation of other feature sets, such as spectrograms, the use of ensemble learning models for increased precision, and improvements in processing speed for wider implementation. This study emphasizes the value of advanced, hybrid machine learning techniques in thwarting the increasing ubiquity of deepfake technology and promoting safe and reliable audio communications.

8.2 Future Work:

Future work on this project will focus on adapting the model to detect emerging deepfake techniques, improving real-time detection by optimizing computational efficiency, and expanding its generalization across diverse languages and acoustic environments. Additionally, data augmentation and adversarial training could strengthen robustness, while integrating audio with video or text for cross-modal deepfake detection would improve accuracy. Enhancing the model's explainability and deploying it in real-world applications such as voice authentication and fraud detection are also key goals. Further, efforts to scale the model for edge devices and optimize it for larger datasets will ensure its broader applicability.

8.3 Applications:

The proposed deepfake audio detection system has several important applications across various industries, particularly in enhancing security and trust in voice-based systems:

- **Voice Authentication and Biometric Security**: The model can be integrated into voice-based authentication systems to detect fraudulent attempts using deepfake audio, improving the security of personal and financial services, such as banking, mobile devices, and voice assistants.

- **Digital Forensics and Media Verification**: The detection system can be used in digital forensics to verify the authenticity of audio recordings in legal investigations, journalism, and media. It helps detect manipulated audio in news reports, interviews, or legal proceedings to ensure the integrity of information.

- **Fraud Prevention**: In sectors such as finance and customer service, the system can be used to prevent voice phishing or fraud, where attackers use deep fake technology to impersonate individuals and manipulate voice-based systems like call centers or voice-activated services.

- **Misinformation and Disinformation Combat**: The detection system can help combat the spread of misinformation and disinformation, particularly in social media platforms, by identifying manipulated audio in videos, podcasts, or viral audio clips that could influence public opinion.

- **Telecommunication and Call Centers**: In customer service industries, the system can be used to safeguard against audio deepfakes in call center environments, ensuring that conversations are not manipulated for fraudulent purposes.

References:

[1] https://www.kaggle.com/datasets/mathurinache/the-lj-speech-data set

[2] T. M. Wani, S. A. A. Qadri, D. Comminiello, and I. Amerini, "Detecting Audio Deepfakes: Integrating CNN and BiLSTM with Multi-Feature Concatenation," Proceedings of the 2024 ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec '24), Association for Computing Machinery, New York, NY, USA, pp. 271–276, 2024. doi: 10.1145/3658664.3659647.

[3] O. A. Shaaban, R. Yildirim and A. A. Alguttar, "Audio Deepfake Approaches," in IEEE Access, vol. 11, pp. 132652-132682, 2023, doi: 10.1109/ACCESS.2023.3333866.

[4] L. Pham, P. Lam, T. Nguyen, H. Nguyen and A. Schindler, "Deepfake Audio Detection Using Spectrogram-based Feature and Ensemble of Deep Learning Models," 2024 IEEE 5th International Symposium on the Internet of Sounds (IS2), Erlangen, Germany, 2024, pp. 1-5, doi:10.1109/IS262782.2024.10704095.

[5] L. Huang and C. -M. Pun, "Audio Replay Spoof Attack Detection by Joint Segment-Based Linear Filter Bank Feature Extraction and Attention-Enhanced DenseNet-BiLSTM Network," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 28, pp. 1813-1825, 2020, doi: 10.1109/TASLP.2020.2998870.

[6] Z. M. Almutairi and H. Elgibreen, "Detecting Fake Audio of Arabic Speakers Using SelfSupervised Deep Learning," in IEEE Access, vol. 11, pp. 72134-72147, 2023, doi:10.1109/ACCESS.2023.3286864.

[7] N. Wilkinson and T. Niesler, "A Hybrid CNN-BiLSTM Voice Activity Detector," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 2021, pp. 6803-6807, doi:10.1109/ICASSP39728.2021.9415081.

[8] R. K. Bhukya, A. Raj and D. N. Raja, "Audio Deepfakes: Feature Extraction and Model Evaluation for Detection," 2024 5th International Conference for Emerging Technology (INCET), Belgaum, India, 2024, pp. 1-6, doi: 10.1109/INCET61516.2024.10593405.

[9] R. Mubarak, T. Alsboui, O. Alshaikh, I. Inuwa-Dutse, S. Khan and S. Parkinson, "A Survey on the Detection and Impacts of Deepfakes in Visual, Audio, and Textual Formats," in IEEE Access, vol. 11, pp. 144497-144529, 2023, doi: 10.1109/ACCESS.2023.3344653

[10] M. Rabhi, S. Bakiras, and R. Di Pietro, "Audio-deepfake detection: Adversarial attacks and countermeasures," Expert Systems with Applications, vol. 250, 2024, Art. no. 123941. [Online]. Available: https://doi.org/10.1016/j.eswa.2024.12394