IDENTITY IMPERSONATION DETECTION MODEL

Using Voice Analysis

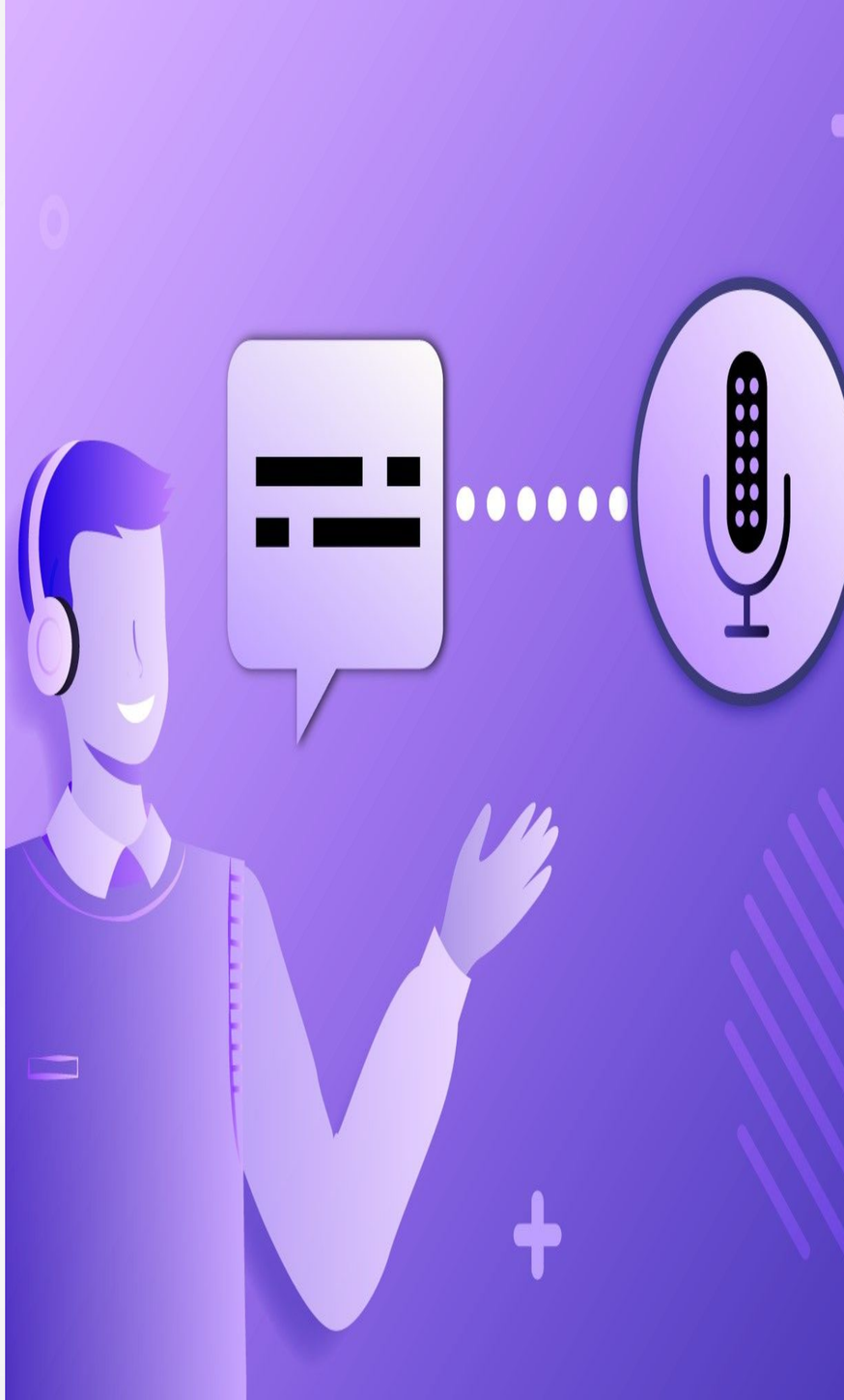
Team: Aster

Team Members:

- Adabala Renukapriya
- Ayan Jain
- Ayush Bhatt
- Mrithulla Padmanabhan
- Saurav Pandey

INTRODUCTION

The importance of identity verification is paramount in today's digital world. Voice analysis offers a comprehensive solution to enhance security and authentication. This presentation will explore the potential of voice analysis in advancing identity verification processes. Comprehensive voice analysis can address these challenges by providing a unique and reliable biometric identifier.

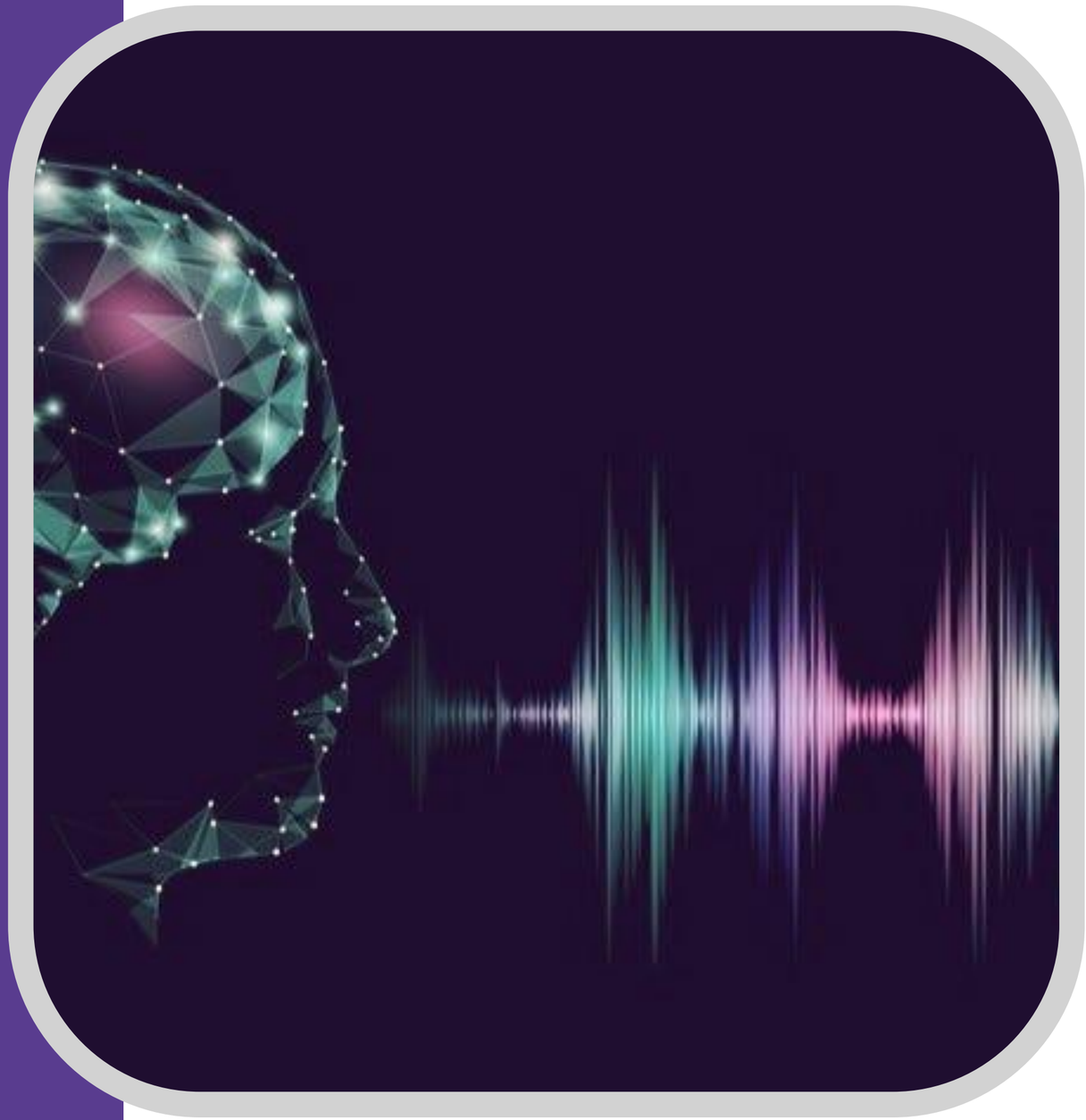


VOICE BIOMETRICS

Voice biometrics utilizes unique vocal characteristics to create a voiceprint. This includes factors such as:

- pitch
- rhythm
- intonation.

The accuracy and reliability of voice biometrics make it an ideal tool for identity verification.



KEY OUTPUT METRICS OF IDENTITY IMPERSONATION DETECTION MODEL

- Checking any voice element is present or not
- Determining is it AI or human
- Language Detection
- Background Noise Detection
- Emotion analysis of Voice

DETECTING SPEECH EXISTENCE

Our project uses Google’s Speech Recognition API to detect if the uploaded audio has any speech content in it.

DETECTING HUMAN vs AI VOICE

Deepfake audio detection is a crucial task in the era of synthetic media, aimed at identifying manipulated or synthesized audio recordings.

In the banking domain, identity personification could be used to impersonate customers and gain access to their accounts. This could lead to financial losses for customers and banks. Banks need to be able to protect their customers from identity theft and fraud. One way to do this is to develop solutions that can detect and prevent AI-generated voice impersonation.

Deepfake Audio Recognition using HMM

- In our project, we extract relevant acoustic features from voice samples, such as Mel-frequency cepstral coefficients (MFCCs), pitch, and intensity. These features are then used as inputs to train the HMM for distinguishing between real and deep fake voices.

- HMM is particularly well-suited for detecting deep fake voices due to its capability to model high-dimensional feature spaces effectively.

Deepfake Audio Recognition using HMM

- It is a classification based ML model which categorizes all the features of a particular voice model and checks for best fit among the test features.

Steps :

1. Extract features using mfcc algorithm. We will extract the first 50 features as they contain relevant audio data for our use case. We will extract these features for all the training dataset. Then for each audio file we will also calculate mfcc_delta and mfcc_delta2.

Deepfake Audio Recognition using HMM

2. After this we pass this through our hmm model for each human and ai voice respectively.(Human -> human_model & AI -> ai_model).

3. For predicting if the audio file is AI or Human, we extract the features of the test audio file and score them against both ai & human models which were created during model training. And the model which gives highest score is the one which our ML model predicts as the speaker.

Deepfake Audio Recognition using CNN

- A convolutional neural network (CNN) is a deep neural network model that is often used for deep fake detection and most commonly applied to analyze visual imagery. We have employed CNNs for the task of deepfake audio detection.

- We experimented with different architectures for CNN model and trained models across various combinations of the training datasets.

Deepfake Audio Recognition using CNN - Feature Extraction

The feature extraction process involves transforming raw audio data into a format suitable for input into the CNN model. **Librosa** library is utilized to extract **Mel-Frequency Cepstral Coefficients** (MFCCs), a common feature representation for audio data.

Steps:

- **Loading audio**: Raw audio data is loaded with a sampling rate of 16000 Hz
- **MFCC Extraction**: MFCCs are computed from the audio signal with parameters specifying the sampling rate and number of MFCC coefficients (40 in our case).
- **Padding or Trimming**: Ensure uniform length of the MFCC matrix. Pad with zeros if frames are insufficient, trim if excessive

Deepfake Audio Recognition using CNN - Model Architecture

Model Architecture:

Convolutional Layers: The convolutional layers with filter sizes and depths for feature extraction.

Max Pooling Layers: Max pooling layers with varying pool sizes to downsample the feature maps.

Batch Normalization: Applied after each convolutional layer to normalize the activations and accelerate training.

Flatten Layer: Flattens the output from the convolutional layers into a 1D tensor for input to the dense layers.

Dense Layers: Fully connected dense layers with ReLU activation for classification.

Dropout Layer: Dropout layer with a dropout rate of 0.5 to prevent overfitting.

Output Layer: Single dense layer with sigmoid activation for binary classification (fake or genuine audio).

Deepfake Audio Recognition using CNN - Model Architecture

Architecture_1:

Sequential model comprising convolutional and pooling layers followed by fully connected layers.

- Input shape: (40, 500, 1) representing 40 MFCC features over 500 frames.
- Convolutional layers:
  - First Conv2D layer with 32 filters of size (3, 3) and ReLU activation.
  - Second Conv2D layer with 64 filters of size (3, 3) and ReLU activation.
  - MaxPooling2D layers with pool size (2, 2) for downsampling.
- Flatten layer to convert 2D feature maps into a 1D vector.
- Dense layers:
  - First dense layer with 128 neurons and ReLU activation.
  - Dropout layer with a dropout rate of 0.5 for regularization.
- Output layer with 1 neuron and sigmoid activation for binary classification.

Deepfake Audio Recognition using CNN - Model Architecture

Architecture_2:

Sequential model comprising convolutional and pooling layers followed by fully connected layers.

- Input shape: (40, 500, 1) representing 40 MFCC features over 500 frames.
- Convolutional layers:
  - Three Conv2D layers with 32 filters of size (3, 3) and ReLU activation.
  - Each Conv2D layer is followed by a MaxPooling2D layer with pool size (3, 3) and strides (2, 2).
  - BatchNormalization layers are applied after the first two Conv2D layers for normalization.
- Flatten layer to convert 2D feature maps into a 1D vector.
- Dense layers:
  - First dense layer with 128 neurons and ReLU activation.
  - Dropout layer with a dropout rate of 0.5 for regularization.
- Output layer with 1 neuron and sigmoid activation for binary classification.

Deepfake Audio Recognition using CNN - Training Data

Various training datasets were taken and we experimented across different combinations to train our models. The datasets we used are:

1. ASVspoof 2019 Dataset: This is a database used for the Third Automatic Speaker Verification Spoofing and Countermeasuers Challenge. Includes human and AI audios spanned across both the genders.
2. Audio Dataset with 10 Indian Languages: This is a massive dataset of human audio samples of 10 different Indian languages. This dataset was created using regional videos available on YouTube
3. DEEP-VOICE: DeepFake Voice Recognition: This dataset contains examples of human speech, and AI versions of those speeches by using Retrieval-based Voice Conversion.

Deepfake Audio Recognition using CNN - Training Data

4. Speaker Recognition Audio Dataset: Dataset consists of 50 speakers' audio recordings, each exceeding 1 hour. The data is converted to WAV format, with a sampling rate of 16kHz and mono channel. It is segmented into 1-minute chunks.

5.WaveFake: This is a novel data set, containing of 117,985 AI generated audio clips from five different network architectures, spanning two languages.

Deepfake Audio Recognition using CNN - Testing Data

Testing data was compiled from various resources, which included different variation from language, accent, gender, noisy, etc.

The models were thoroughly tested on completely unseen and diverse dataset.

Deepfake Audio Recognition using CNN - Model Training and Results

Following models were trained:

	Model-Name	Architecture	Datasets-Used	Accuracy(%)		
				AI	Human	Overall
1	Model_1	Architecture_1	1, 3	0.9	0.42	0.66
2	Model_2	Architecture_1	1, 3	0.62	0.46	0.54
3	Model_3	Architecture_1	1, 2, 5	0.98	0.46	0.72
4	Model_4	Architecture_1	4, 5	0.8	0.46	0.63
5	Model_5	Architecture_2	1, 3	0.8	0.2	0.5
6	Model_6	Architecture_2	2, 5	0.8	0.86	0.83
7	Model_7	Architecture_2	1, 2, 5	1	0.1	0.55
8	Model_8	Architecture_2	4, 5	0.66	0.68	0.67

Deepfake Audio Recognition using pre-trained models

We also experimented with open-source fine-tuned pre-trained models. The models we experimented with are:

1. **wavlm-base-960h-asv19-deepfake**[1]: This model is a fine-tuned version of microsoft/wavlm-base
2. **hubert-base-960h-asv19-deepfake**[2]: This model is a fine-tuned version of facebook/hubert-base-ls960
3. **hubert-base-960h-itw-deepfake**[3]: This model is a fine-tuned version of facebook/hubert-base-ls960
4. **wav2vec2-base-960h-itw-deepfake**[4]: This model is a fine-tuned version of facebook/wav2vec2-base-960h
5. **wavlm-base-960h-itw-deepfake**[5]: This model is a fine-tuned version of microsoft/wavlm-base

Deepfake Audio Recognition using pre-trained models - Results

The following table summarizes results for experiments conducted for pre-trained models:

	Model	Accuracy(%)		
		AI	Human	Overall
1	wav2vec2-base-960h-itw-deepfake	0.18	0.9	0.54
2	hubert-base-960h-asv19-deepfake	0.63	0.24	0.435
3	hubert-base-960h-itw-deepfake	0.49	0.72	0.605
4	wavlm-base-960h-itw-deepfake	0.2	0.82	0.51
5	wavlm-base-960h-asv19-deepfake	0.6	0.1	0.35

EMOTION ANALYSIS OF SPEECH

Speech Emotion Analysis aims to decipher the emotional content conveyed through speech signals. Leveraging Python programming, our project employs a **Multilayer Perceptron (MLP) classifier**, a robust neural network model capable of learning intricate patterns within the data. Through this innovative approach, we harness the power of machine learning to accurately detect and classify 8 emotions: happiness, sadness, anger, neutral,calm, fearful, disgust and surprised from speech samples.

STEPS INVOLVED:

- Dataset used: RAVDESS
- Loading the dataset
- Converting the audio files in dataset to text
- Down Sampling and Masking the audio files in order to remove the interference of background noise
- Visualizing the spectrograms of voice samples
- Feature Extraction
- Splitting the training dataset and training the model
- This model is giving an accuracy of **0.8146**

ADDITIONAL DETECTIONS

Additionally we are predicting:

- Background noise level

  Method: The audio is de noised and the difference between original and de-noised audio is calculated. If that frequency is above a certain threshold then it is categorized accordingly as High, Medium and Low Noise Level

- Language of the voice

  Method: Whisper API base Model has been used to detect the language of the audio clip.