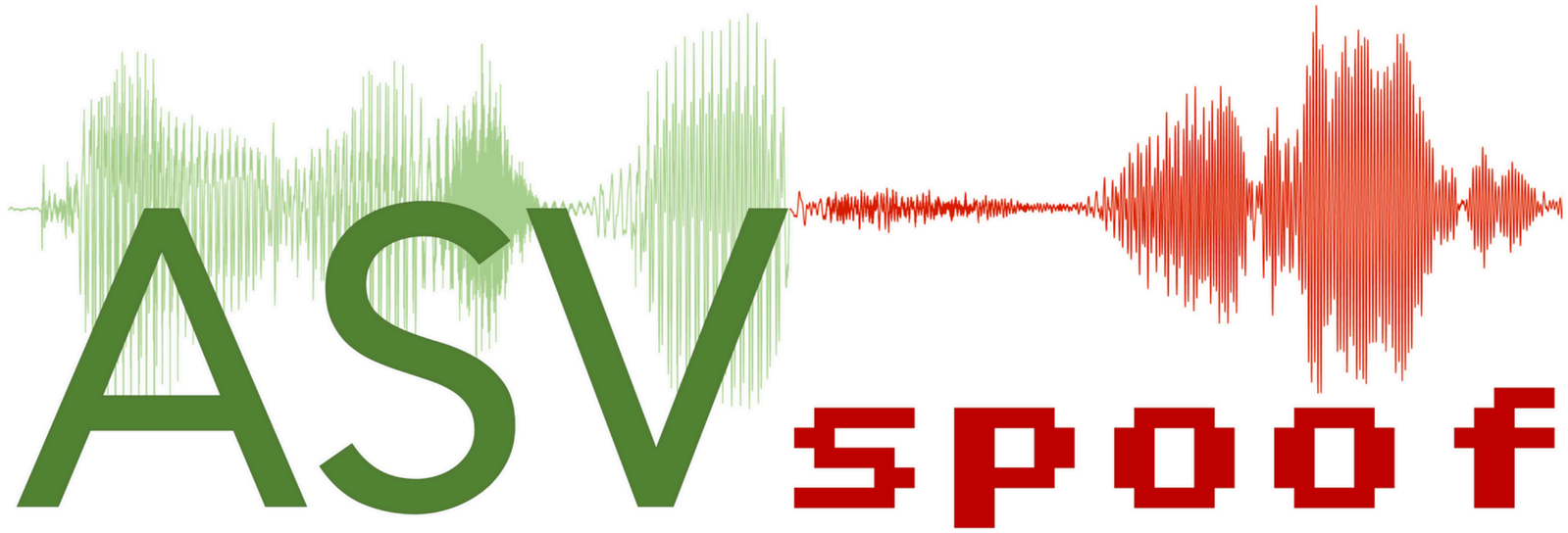
Deepfake Speech Detection



Automatic Speaker Verification and
Spoofing Countermeasures Challenge

Overview

- This work addresses the challenge of distinguishing between genuine (bonafide) and synthetic (spoofed) speech by leveraging advanced pre-trained models like **Wav2Vec2 with Bi-LSTM**, **WavLM**, **HuBERT** and **ResNet**

- It capture rich spectral and temporal audio features. Fine-tuned on the **ASVspoof 2019 dataset**, the proposed binary classification framework effectively detects deepfake audio, demonstrating high accuracy, robustness

Proponents

Ayush Singh(22bds012)

Hubert Model

Nachiket Apte(22bds041)

WavLM Model

Yashraj Kadam(22bds066)

BiLSTM Model, Dashboard(Streamlit)

Harsh Raj(22bds027)

Residual Networks Model

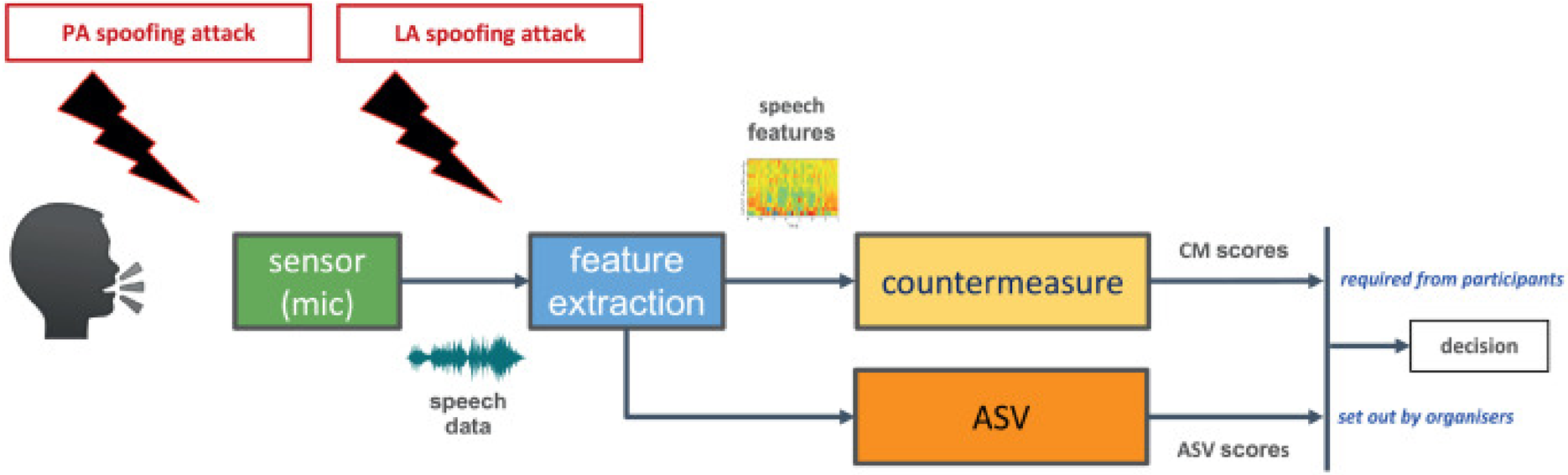
About the Dataset

This is a database used for the Third Automatic Speaker Verification Spoofing and Countermeasures Challenge, for short, ASVspoof 2019. The ASVspoof 2019 database is designed to reflect two different use case scenarios, namely logical and physical access control. For this experiment, only the logical access control part is used. Logical access (LA) control implies a scenario in which a remote user seeks access to a system or service protected by ASV.In the LA scenario, it is assumed that spoofing attacks are presented to the ASV system in a worst-case, post-sensor scenario. Attacks then take the form of synthetic speech or converted voice, which are presented to the ASV system without convolutive acoustic propagation or microphone effects. ASV/CM training data comprises 2,580 bonafide utterances and 22,800 spoofed utterances generated by using four TTS and two VC algorithms. The ASV/CM development partition contains 1,484 bona fide target utterances, 1,064 bona fide non-target utterances, and 22,296 spoofed utterances generated with the same TTS and VC algorithms.

About the Dataset

The Dataset is divided into two parts:

1. Logical Access
2. Physical Access



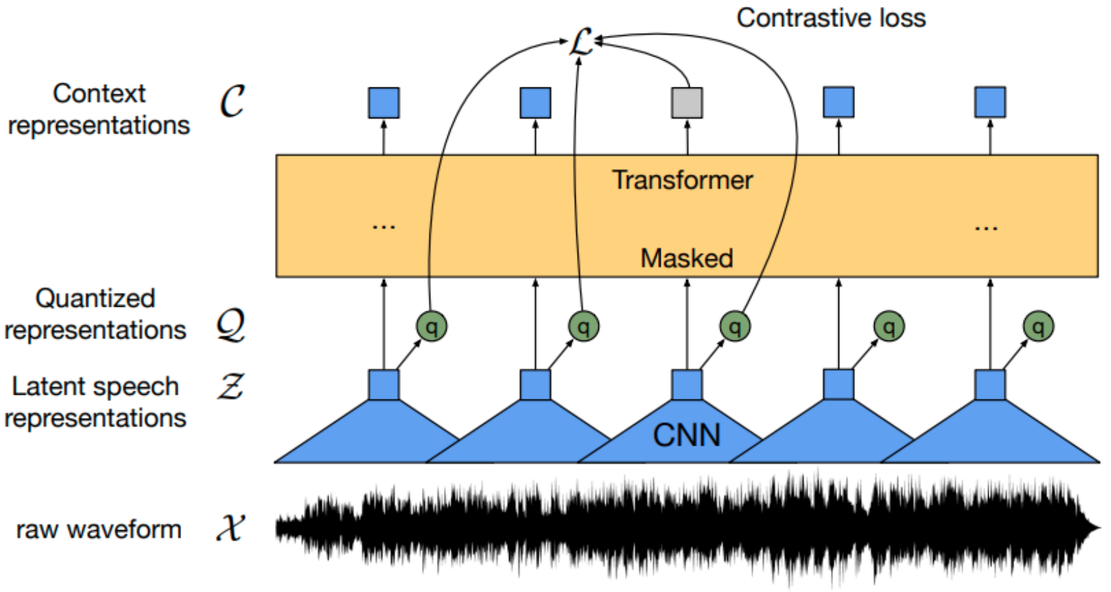
For this experiment, we are considering only the Logical Access part

Wav2Vec

Wav2vec2 Architecture: The complete architecture of the framework can be divided into 3 components:

Feature encoder: This is the encoder part of the model. It takes the raw audio data as input and outputs feature vectors. Input size is limited to 400 samples which is 20ms for 16kHz sample rate. The raw audio is first standardized to have zero mean and unit variance. Then it is passed to 1D convolutional neural network (temporal convolution) followed by layer normalization and GELU activation function. There could be 7 such convolution blocks with constant channel size (512), decreasing kernel width (10, 3x4, 2x2) and stride (5, 2x6). The output is a list of feature vectors each with 512 dimensions.

Transformers: The output of the feature encoder is passed on to a transformer layer. One differentiator is use of relative positional embedding by using convolution layers, rather than using fixed positional encoding as done in the original Transformers paper. The block size differs, as 12 transformer blocks with model dimension of 768 are used in BASE model but 24 blocks with 1024 dimension in LARGE version.



Quantization module: For self-supervised learning, we need to work with discrete outputs. For this, there is a quantization module that converts the continuous vector output to discrete representations, and on top of it, it automatically learns the discrete speech units. This is done by maintaining multiple codebooks/groups (320 in size) and the units sampled from each codebook are later concatenated (320x320=102400 possible speech units). The sampling is done using Gumbel-Softmax which is like argmax but differentiable.

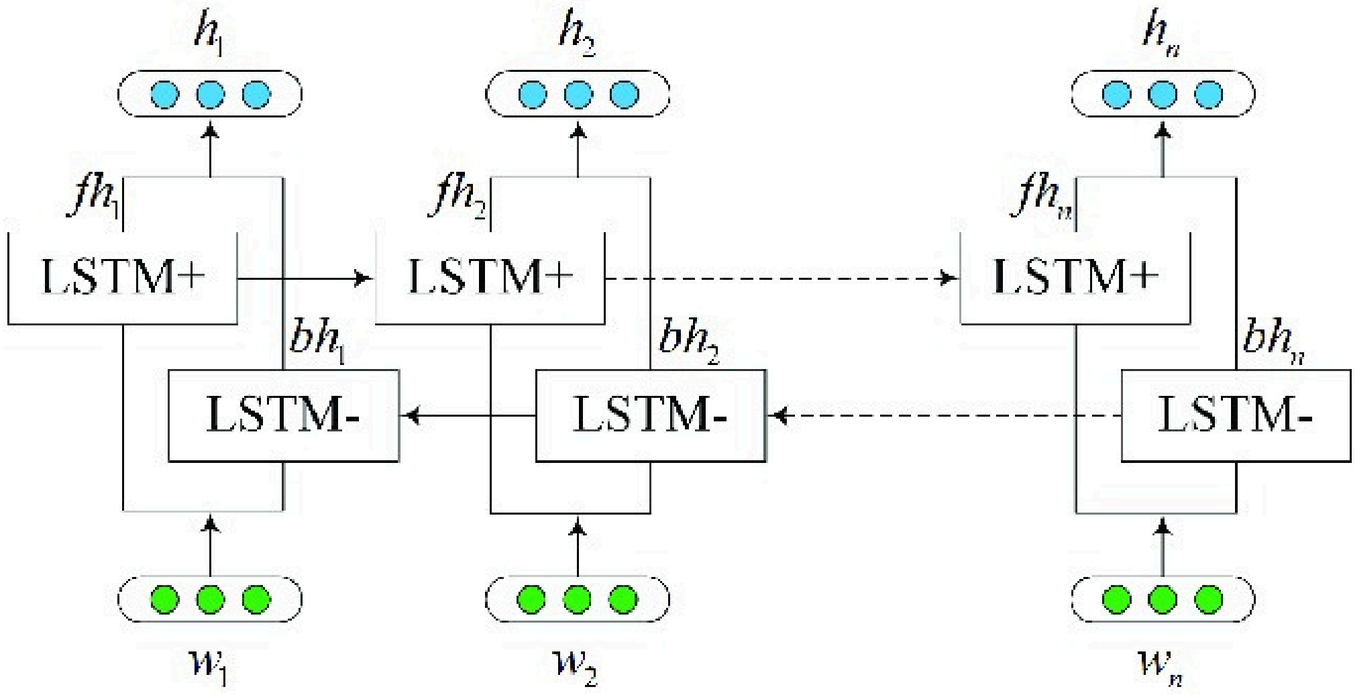
wav2vec2 + Bi-LSTM

Model Architecture

Feature Extraction with Wav2Vec2: Wav2Vec2 extracts contextualized audio embeddings directly from raw signals, providing rich representations that are fed into the Bi-LSTM network.

Bi-LSTM Network: A Bidirectional LSTM layer models sequential dependencies in the extracted features by processing them in both forward and backward directions.

Classification Head: The Bi-LSTM outputs are passed through: Adaptive Average Pooling to reduce the dimensionality and aggregate key features across the sequence.A fully connected linear layer for further dimensionality reduction.A sigmoid activation function to produce a binary classification output (spoof or bonafide).



Training and Optimization

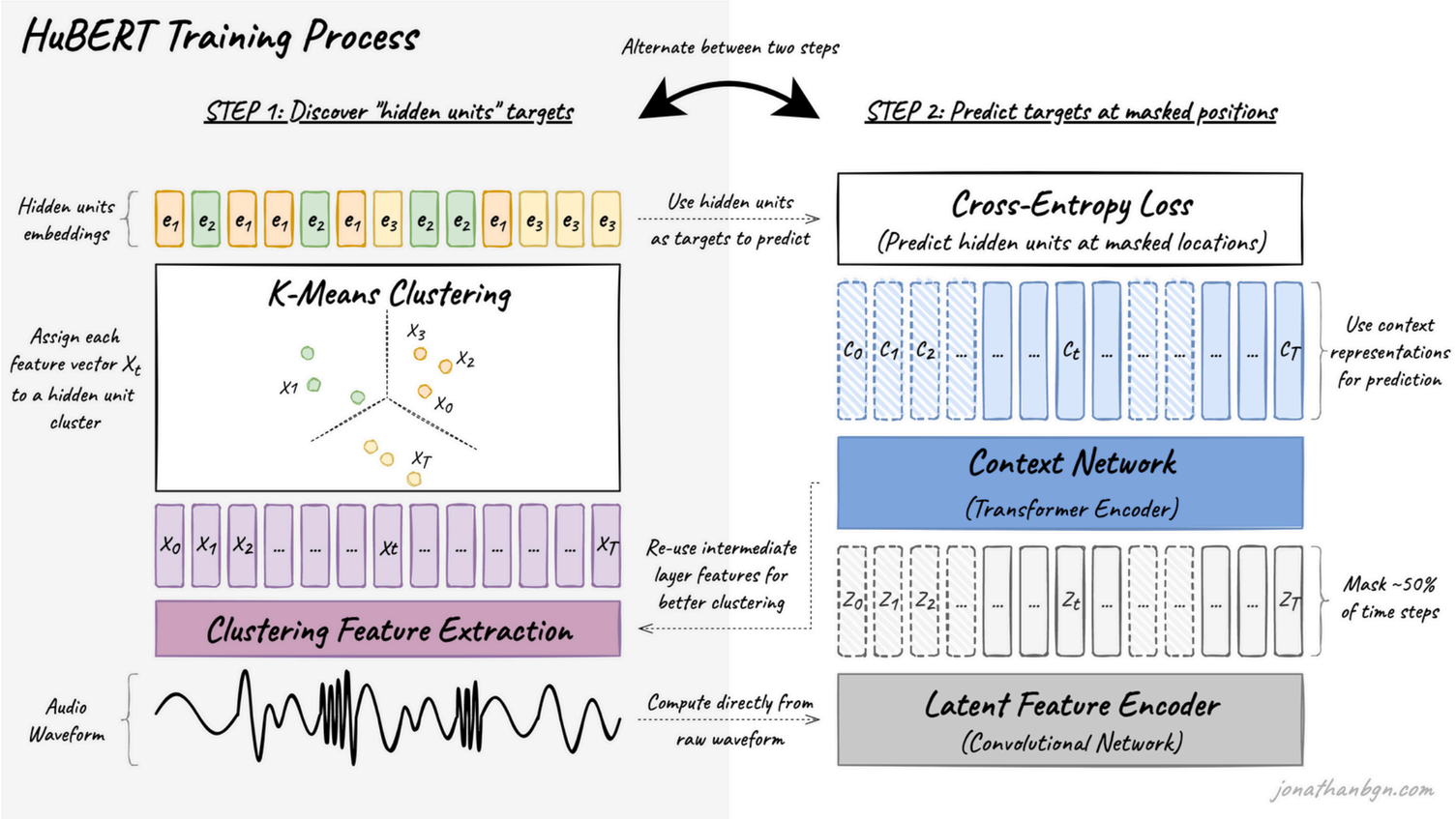
The model is trained using Binary Cross-Entropy Loss for binary classification, with the Adam optimizer (learning rate=1e-5) to ensure efficient and stable convergence. Regularization techniques include dropout layers within the Bi-LSTM to prevent overfitting and early stopping to terminate training when validation performance plateaus.

Hubert model

Model Architecture

Transformer Encoder: The HuBERT model employs a transformer-based encoder to process the extracted audio features. By leveraging its multi-head self-attention mechanism, the encoder captures complex temporal dependencies and patterns in sequential audio data, enabling the model to distinguish subtle variations between spoofed and genuine speech.

Classification Layer: The encoder's output embeddings are passed through a fully connected classification layer with a softmax activation function. This layer outputs the probability scores for the two target classes: spoof and bonafide.



Training and Optimization

The model is fine-tuned using Cross-Entropy Loss to minimize prediction errors and the Adam optimizer with a learning rate of 1e-5 for stable and efficient convergence. Regularization strategies include using a small batch size for better generalization and early stopping to mitigate overfitting.

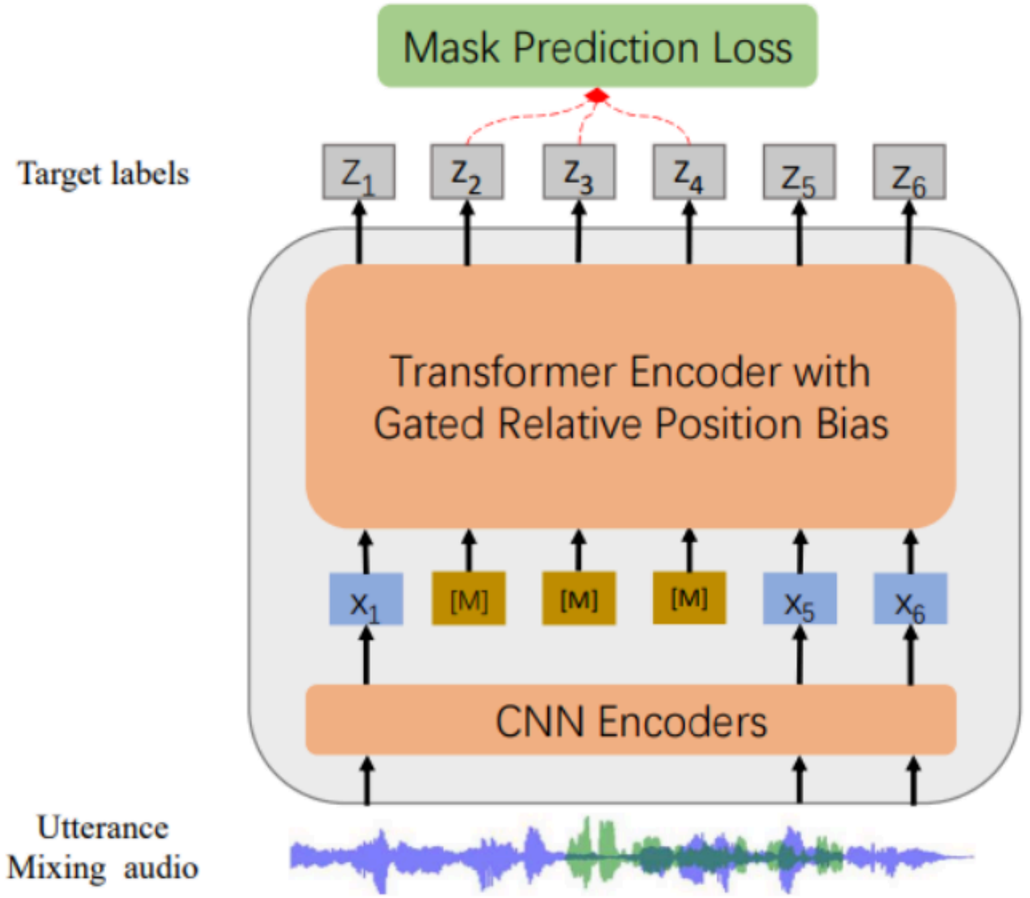
WavLM model

Model Architecture

Pre-Trained WavLM Model: The extracted features are passed through the last hidden layer of the pre-trained WavLM model. This layer encodes the input features into high-dimensional representations, effectively capturing both local and global patterns in the audio data.

Feature Aggregation: To reduce the dimensionality and aggregate the learned representations, an adaptive average pooling layer with an output size of 128 is applied. This ensures that the feature size is consistent, regardless of input audio length.

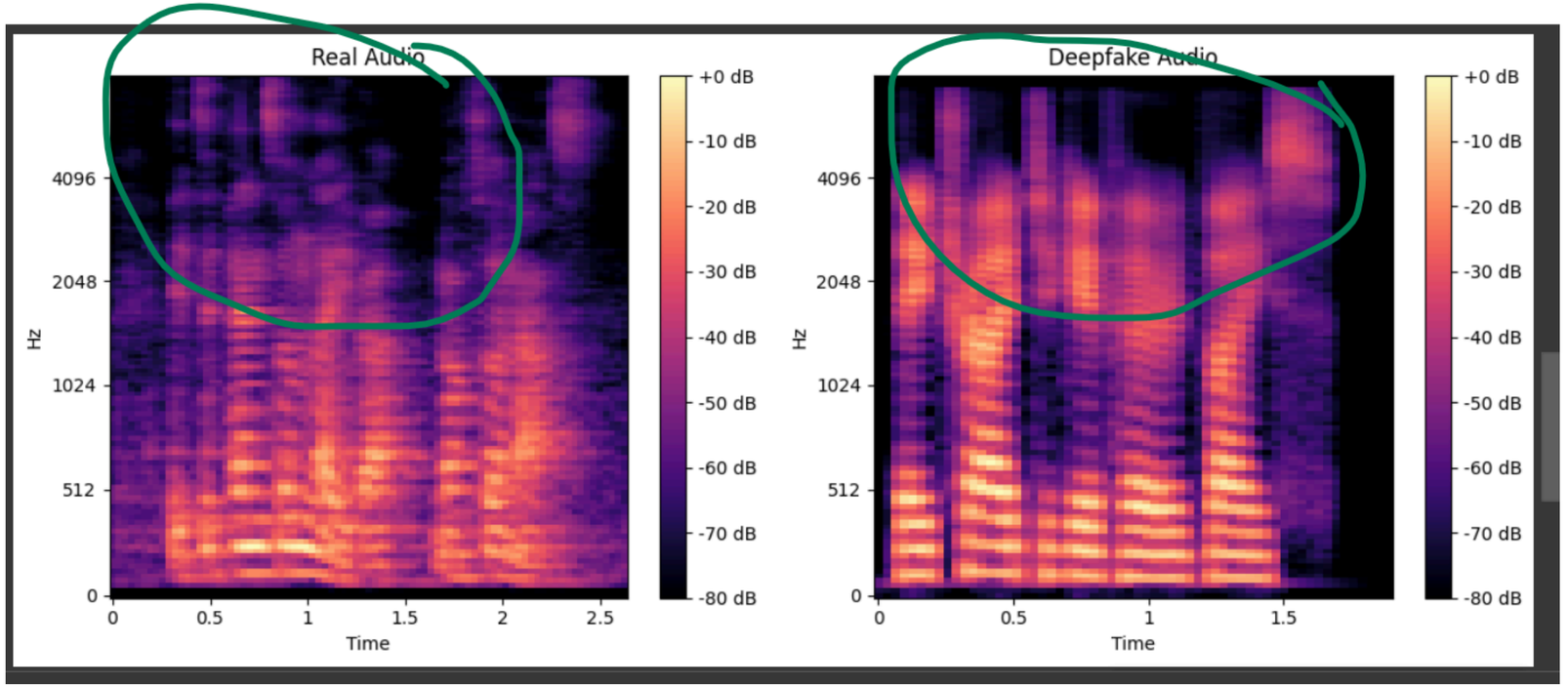
Classification Head: The pooled features are passed through a linear layer, which maps the aggregated features to a lower-dimensional space. A sigmoid activation function, producing a binary output for classification into spoof or bonafide categories.



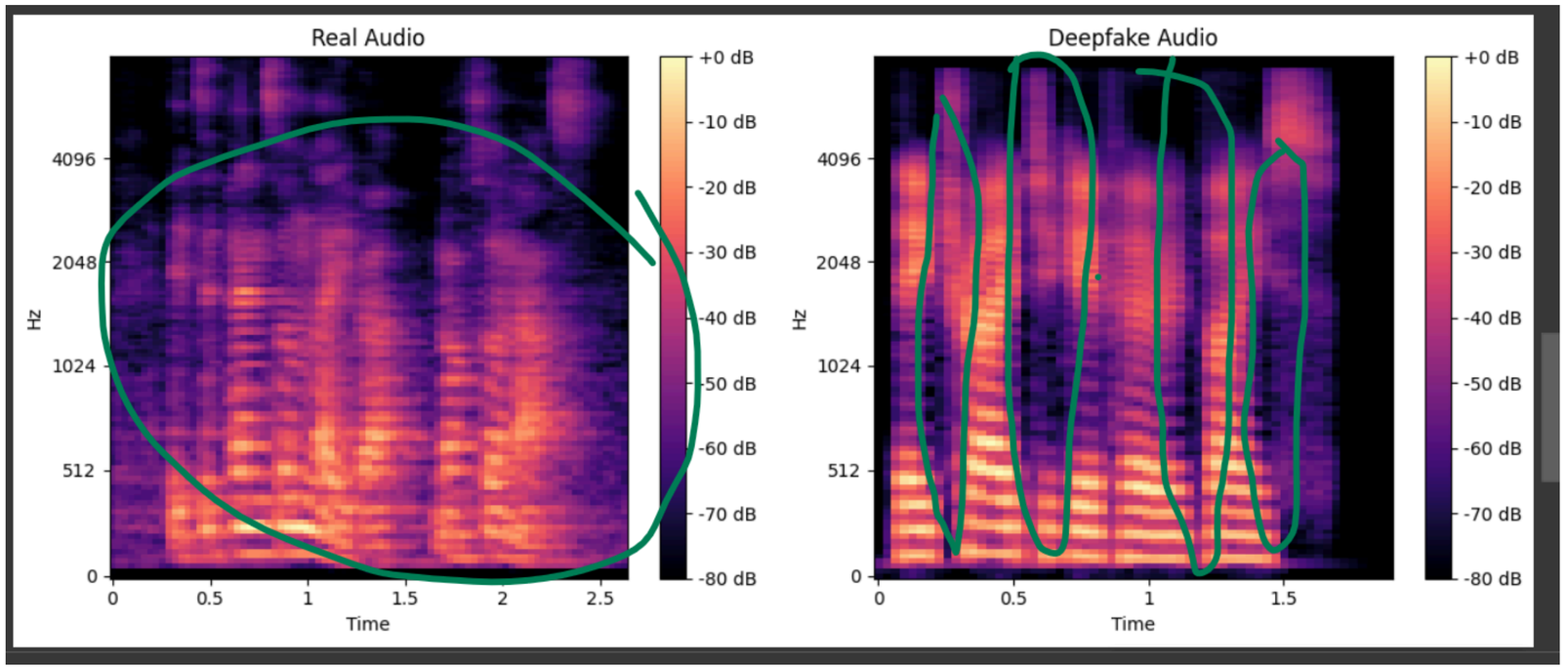
Training and Optimization

The model is fine-tuned using Binary Cross-Entropy Loss to minimize classification error, with the Adam optimizer ensuring efficient and stable gradient updates. Adaptive learning rate scheduling facilitates optimal convergence, and early stopping prevents overfitting while promoting generalization

Features of Spectrogram:



Spectral Smoothing: Deepfake audio may show an over-smooth appearance in the spectrogram due to the synthetic nature of the audio generation process.

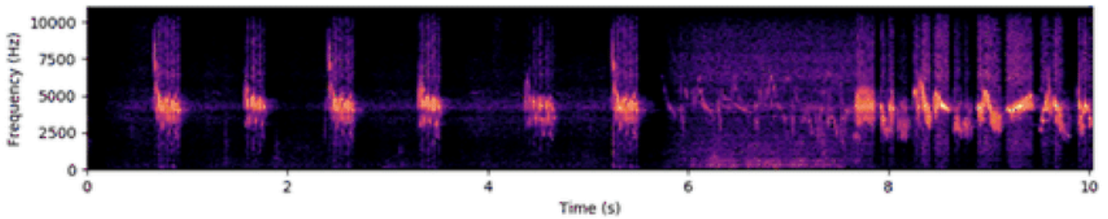


Unusual or consistent background noise that doesn't align with human speech patterns can be a sign of synthetic audio

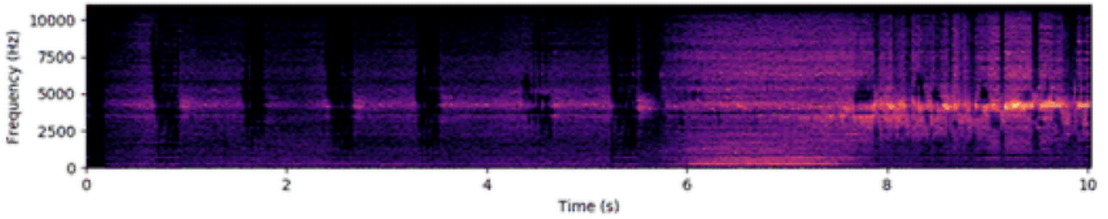
Harmonic Distortion:

Real Audio: In real audio, harmonics appear as regular, evenly spaced horizontal lines or bands in a Mel spectrogram. For voiced sounds, such as human speech, these lines are often clear and consistent.

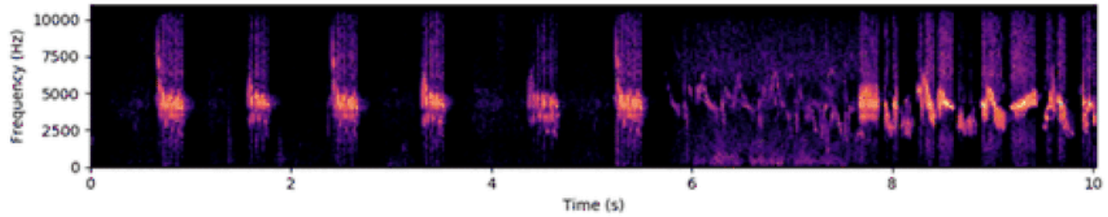
Deepfake Audio: Deepfake audio might show irregularities in these patterns. You may notice uneven spacing, extra or missing harmonics, or irregularly shaped harmonic structures due to distortion introduced by the synthesis process.



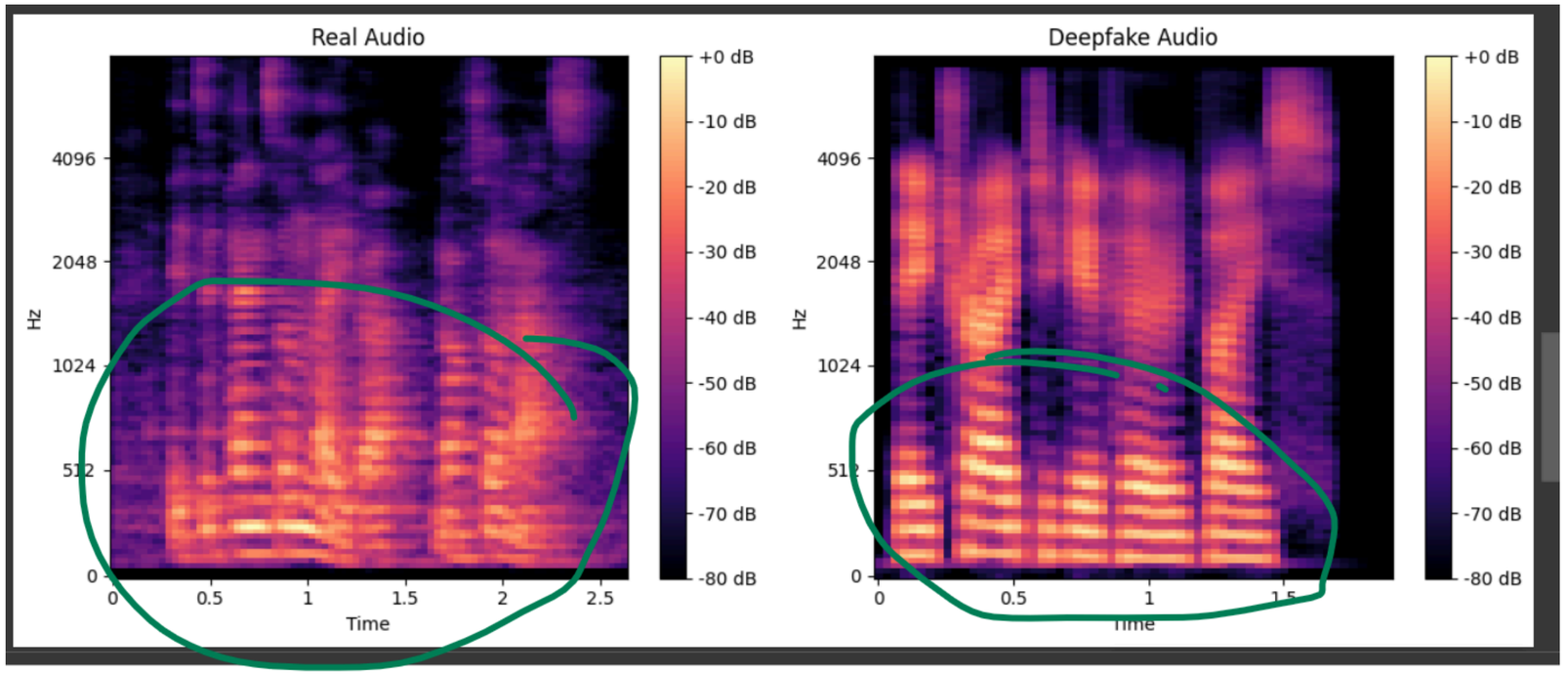
(a) Original spectrogram.



(b) Harmonic spectrogram.



(c) Percussive spectrogram.

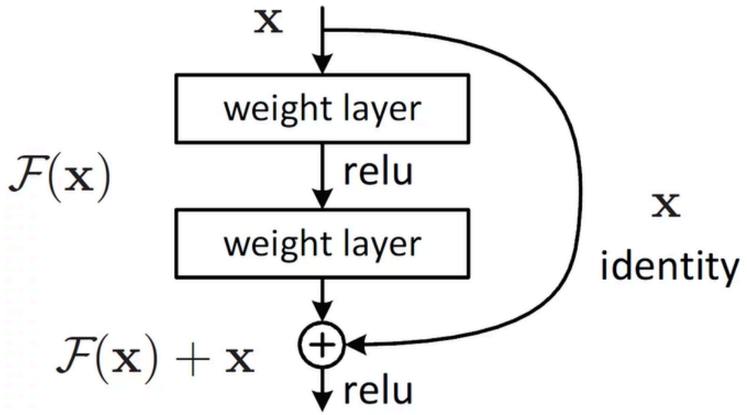


Unusual Energy Concentration: Notice if certain frequency bands have unusually high or low energy compared to others. For example, energy might be concentrated in unusual frequency ranges or show non-standard distributions.

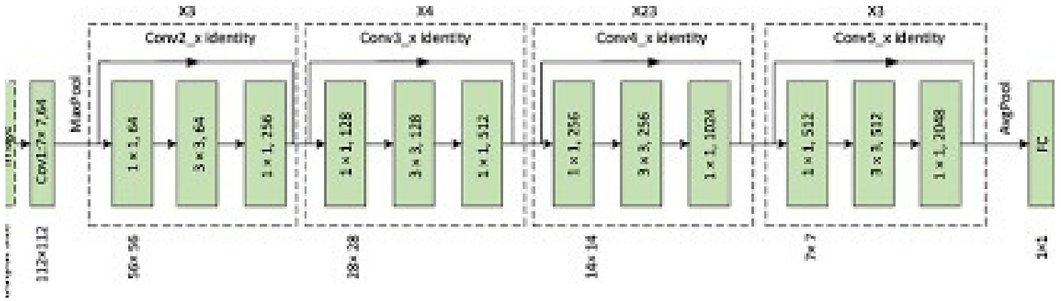
ResNet model

Model Architecture

Pre-Trained ResNet101: The model leverages a pretrained ResNet101 architecture, with modifications to adapt it for single-channel input:
The first convolutional layer is modified to accept one input channel. The initial 39 layers of the model are frozen to retain pre-trained features while reducing computational load.



Feature Extraction: The ResNet's feature extractor outputs are passed through a custom head comprising Adaptive Average Pooling (aggregates features), Flattening (prepares for dense layers), a Fully Connected Layer (maps to a single output), and a Sigmoid Activation (produces binary classification probabilities for spoof or bonafide).

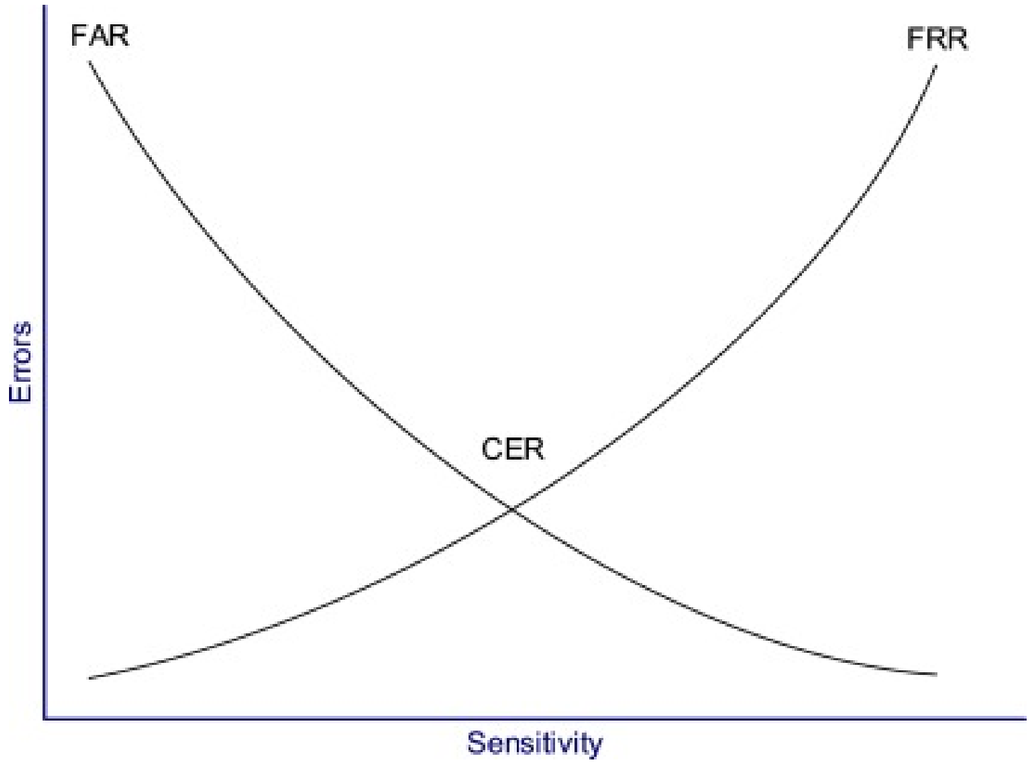


Training and Optimization

The model is trained with Binary Cross-Entropy Loss for classification, using the Adam optimizer for efficient convergence. A learning rate scheduler dynamically adjusts the learning rate during training, and dropout in the custom layers helps prevent overfitting.

Results

The models have given significantly great results on the dataset with very low EER/CER . EER stands for Equal Error Rate. It is the intersection point of the curves of False Acceptance Rate and False Rejection Rate in the plot of error vs sensitivity. This intersection point ensures that the model is unbiased towards False Positives and False Negatives. The low EER is an indication of how strong the model is in classifying the data as spoofed or bonafide.



	Models			
	HuBERT	WavLM	Bi-LSTM	ResNet
EER	0.36	0.01	0.45	0.30

yashraj9922/
echo.ai

1 Contributor | 0 Issues | 0 Stars | 0 Forks

yashraj9922/echo.ai

Contribute to yashraj9922/echo.ai development by creating an account on GitHub.

GitHub

Thank You