Recognition of Audio Source Recording Device using MFCC and RNN

[*1]Venkata Lalitha Narla, [2]B.Harsha Vardhini, [3]N.S.S.S.Kavitha, [4]P.Ashritha, [5]M.Geetha
[1,2,3,4,5]Department of Electronics and Communication Engineering
Aditya College of Engineering & Technology, Surampalem, AP, India
*lalithanarla.ece@gmail.com

Abstract: Accurate identification of audio source recording devices is paramount in digital forensic investigations, including topics like copyright protection, tamper detection, and audio source forensics. This work presented a novel method for learning feature representations using temporal audio characteristics, such as Mel Frequency Cepstral Coefficients (MFCC) and Constant-Q Transform (CQT), obtained from segmented acoustic features. Subsequently creates a structured representation learning model by combining Long Short-Term Memory Networks (LSTM) with Recurrent Neural Networks (RNN). This model efficiently condenses spatial information, resulting in accurate recognition, by utilizing temporal modelling and time-frequency representation. The performance of the proposed methods is tested on 10-second audio signals recorded with four different audio recording devices. The outcomes of the experiment show an amazing degree of accuracy with 96% in classifying four types of recording audio source devices. This method promises improved efficacy in a variety of forensic circumstances and represents a substantial development in audio forensic analysis. The performance metrics of audio source recording using CQT-RNN and MFCC-RNN are compared, and also compared with state-of-the-art methods. A user interface has been developed to facilitate the recognition of the source device for test audio signals using the proposed method. Overall, this research marks a substantial advancement in audio forensic analysis, providing a robust, accurate, and user-friendly solution for the identification of audio source recording devices, and underscoring its potential for widespread forensic applications.

Keywords: Digital Forensics, Audio Source Recording Device, Constant-Q Transform, Mel Frequency Cepstral Coefficients, Recurrent Neural Networks, Long Short-Term Memory Networks.

I. Introduction

In the field of digital forensics, identifying Audio Source Recording (ASR) devices is critical, acting as a foundation for many investigative techniques. From determining the origin of audio evidence in judicial processes to detecting instances of tampering or infringement, the ability to reliably trace recordings to specific devices is critical [1]. Despite their importance, present approaches for ASR device recognition frequently face limits due to insufficient information utilization, resulting in inferior accuracy in real-world applications [2].

This study aims to address these issues by introducing a unique approach based on feature representation learning. The methodology attempts to overcome the problems that have limited the usefulness of current methods by using the natural properties of audio signals and modern machine-learning techniques. The technique relies heavily on temporal audio features, notably the Constant-Q Transform (CQT) and Mel Frequency Cepstral Coefficients (MFCC), which provide detailed representations of audio content over time. The addition of temporal segmentation increases the granularity of the feature extraction technique, allowing for a more nuanced analysis of audio data.

Building upon these fundamental components, it proposes integrating Recurrent Neural Network (RNN) and Long Short-Term Memory Network (LSTM) to create an organized representation learning model. This hybrid architecture takes advantage of the characteristics of both RNNs and LSTMs, leveraging their temporal modeling skills to efficiently record and evaluate the sequential nature of audio signals. One of the fundamental features of the methodology is its capacity to condense spatial information using time-frequency representation, allowing for more precise device recognition. By incorporating temporal dynamics and frequency-domain properties, the approach overcomes the constraints of existing feature extraction methods, providing a comprehensive framework for audio forensics investigation.

The remaining article is as follows: a survey of the state-of-art works are discussed in Section II. The proposed ASR device identification process using feature extraction and RNN training is presented in Section III. Experimental results and conclusion are explored in Section IV and Section V respectively.

II. Literature Survey

Intricacies of the experimental framework, findings and implications of state-of-art works are discussed in this section. Through this detailed examination, seek to demonstrate the approach's transformative potential and implications for the field of digital audio forensics.

Multimedia forensics involves determining the source device of a signal. R.Buchholz et.al. [3] proposed a scheme to determine the model of the audio device involved in the recording of audio samples. A Fourier coefficient histogram with two different FFT window sizes for nine different thresholds for approximately silence segments of the DAR is obtained and considered as a feature vector. The authors used WEKA machine-learning tool for classification. Naïve Bayes, SMO, Simple Logistic, J48, IB1 and IBk are the selected classification algorithms from WEKA. A simple Logistic classifier has achieved a classification accuracy of 93%.

H. Malik and J W Miller [4] classified microphones based on microphone-induced artifacts that are modeled using a non-linear function. A statistical method based on bi-coherence is then used to capture microphone-induced nonlinearity. A similarity measure based on fourth and lower-order statistics of frame-based scale invariant Hu moments is used for microphone classification. The effectiveness of this scheme is tested with 24 audio recordings captured using eight microphones during three recording sessions and evaluated using ambient noise recording only.

Source cell phone microphone recognition using non-voice segments of DAR is proposed by C.Hanilci and T. Kinnunen [5]. Two datasets viz., TIMIT and Live records are considered for experimentation using classifiers viz., SVM, GMM-ML and GMM-MMI. Features namely MFCC and LFCC are explored in non-voice segments. It is concluded that extracting features using non-voice portions resulted in higher recognition rates when compared with features from voiced segments.

O.Eskidere explored LPCC, PLPC and MFCC to obtain features, and a Gaussian mixture model was utilized to determine source microphones [6]. This scheme is evaluated on 16 different sources for speaker-dependent and speaker-independent cases. Authors claimed that LPCC features have provided the highest recognition rate.

Piczak examines the application of the use of CNNs for environmental sound classification [7]. Their research shows that CNNs are effective in automatically classifying ambient noises like animal calls, machinery noise, and urban sounds. Using CNNs' hierarchical feature learning capabilities, the suggested approach achieves excellent classification accuracy, setting the framework for advanced audio analysis systems in environmental monitoring and surveillance. Salamon and Bello study the use of deep CNNs and data augmentation approaches to classify environmental sounds [8]. Their findings highlight the effectiveness of CNNs in learning discriminative features from spectrogram representations of ASR. The proposed technique enhances the resilience of the classification model, leading to enhanced performance in real-world audio classification tasks.

Hershey et al. introduce deep clustering, a technique for learning discriminative embedding for audio segmentation and separation problems [9]. Their research uses deep neural networks to develop representations that aid in the clustering of audio segments depending on their content, allowing for tasks such as speaker diarization and source separation. Deep clustering has shown encouraging results in a variety of audio processing applications, indicating its potential to advance the discipline.

Gemmeke et al. introduce AudioSet, an ontology-based and human-labeled collection of audio events [10]. Their research aims to contribute to the development of audio event detection systems by providing a large-scale dataset annotated with distinct audio events. AudioSet has proven to be a useful resource for academics working on audio event detection, enabling them to train and evaluate machine learning models on real-world audio datasets.

The audio signal is the sum of the speech signal and noise signal. Authors S.Q.Z.Huang and Y.L.S.Shi [11] proposed a deep learning approach with noise as an intrinsic feature to identify the source device of DAR. The authors compared Softmax, Multilayer perceptron and Convolutional neural network classifiers and compared parameters in one certain classifier.

G.Baldini & I.Amerini [12] presented a detection and authentication method by exciting smartphone microphones with non-speech portions at different frequencies. A broad database of 32 smartphones was utilized to assess the performance of this method. Authors reported that the CNN employed provided significant identification and authentication accuracy in different operational scenarios and in the presence of Gaussian noise, babble and street noise.

A copy-move forgery detection of speech recording based on the correlation between pitch and formant has been proposed [13]. Here speech is divided into voiced and non-voiced speech segments then pitch and formant sequences are extracted as features for voiced segments. Similarities between formant sequences as well as pitch sequences are calculated with the help of a dynamic time-warping algorithm. Authors used WSJ and TIMIT speech databases to evaluate the performance of post-processing operations viz., white Gaussian noise, pink noise, median filter, MP3 compression, etc., and compared with the state-of-art works.

The performance of Gaussian Supervector (GSV) features in microphone recognition, focuses on the impact of Universal Background Model (UBM) parameters [14]. The raw GSV, containing both microphone and speech information, can be noisy. To improve GSV quality, the authors proposed a kernel-based projection method that maps the raw GSV to a new feature space, aiming to separate microphone and speech information. Experimental results show that the projected GSV consistently

outperforms the raw GSV using linear Support Vector Machine (SVM) and Sparse Representation-based Classifier (SRC), demonstrating the effectiveness of the projection method.

Diego Renza, Jaisson Vargas and Dora M. Ballesteros [15] proposed a scheme to identify manipulations in the audio signal through MFCC, PCA and RSA. Original and manipulated audio hash is compared with the help of BER threshold value and measured the integrity of the content.

The authors presented a deep neural network-based real-time sound source localization (SSL) model designed for low-power IoT devices using microphone arrays [16]. The SSL model processes multi-channel acoustic data through parallel convolutional neural network layers to capture unique delay patterns across frequency ranges, estimating both fine and coarse voice locations. The model achieved 91.41% accuracy in fine location estimation and a 7.43° direction of arrival error on noisy data, with a processing time of 7.811 ms per 40 ms samples on a Raspberry Pi 4B. The model is suitable for camera-based humanoid robots to enhance voice interaction in crowded environments.

ASR devices along with the environment in which audio is recorded are identified [17]. In this work, authors automatically extracted the environment and microphone features using a Convolutional Neural Network and Long-Short Term Memory from the speech signal. The authors conducted experiments and calculated classification accuracy by considering only voiced segments, only unvoiced segments and combined segments. In this investigation, they concluded that using unvoiced segments got good accuracy results compared to voiced segments. In this experimentation, speech datasets which are recorded in three environments and four recording devices are used with different audio quality levels.

ASR device identification based on the speech recordings is presented in [18]. Three fingerprints are extracted viz., channel response, cuccovillo, band energy difference from speech recordings and classified in this work. The performance of the three features is studied individually by observing the classification accuracy.

Multimedia forensics research has explored various techniques for identifying audio source recording devices, leveraging features like Fourier coefficients, MFCC, LFCC, and non-voice segments for improved recognition. Methods include machine learning classifiers such as SVM, GMM, and CNNs, demonstrating high accuracy in diverse scenarios. Recent advancements emphasize deep learning approaches, like deep clustering and real-time sound source localization, enhancing the robustness and applicability of audio forensic analysis. Challenges remain in extracting inherent device characteristics and developing effective recognition models to improve device identification accuracy.

The two main challenges in recording device identification research are:

- Determining the inherent qualities of the source devices and extracting expressive information from recordings.
- Creating effective recognition models is the second step, and it will significantly increase the accuracy of recording device recognition.

The proposed methods ASR-CQT-RNN and ASR-MFCC-RNN will provide appropriate solutions for these problems.

III. Proposed system

The proposed system for identifying audio source recording devices involves creating a diverse dataset from various devices and extracting features using Mel Frequency Cepstral Coefficients (MFCC) and Constant-Q Transform (CQT). These features are then processed using a structured representation learning model that combines Long Short-Term Memory Networks (LSTM) with Recurrent Neural Networks (RNN) to enhance recognition accuracy. A user interface is developed to facilitate real-world application, enabling efficient and accurate identification of audio source devices in forensic investigations. The system's performance is validated through comparative analysis with state-of-the-art methods, ensuring its robustness and reliability.

The generic flow of the proposed work is shown in Figure 1 which involves six steps: Data Collection, Feature Extraction, Model Training, Evaluation, Testing and Deployment.

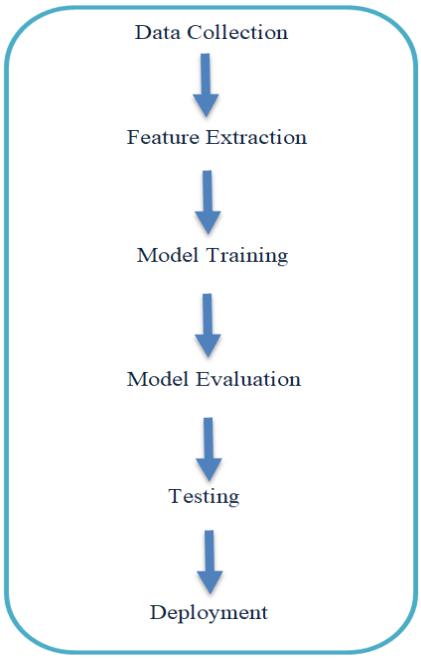


Fig 1: Generic Flow

1) Data collection:

Create a dataset of ASRs from various source devices, guaranteeing diversity in terms of device, recording settings, and content. The expression is shown in Eq (1).

$$S_i^n = \begin{bmatrix} a_1^1 & a_2^1 & \cdots a_i^1 \\ a_1^2 & a_2^2 & \cdots a_i^2 \\ \vdots & \vdots & \vdots \\ a_1^n & a_2^n & \cdots a_i^n \end{bmatrix} \quad (1)$$

Where i indicates the number of samples

n indicates the number of ASR devices based on the dataset

2) Feature Extraction:

Extract useful features from preprocessed audio data. Common time-frequency representations include MFCC, CQT, and others. The feature extraction stage converts raw audio data into a machine-learning

algorithm-compatible format while keeping crucial audio signal characteristics. In the proposed system for enhanced source recording device recognition, compare two prominent feature extraction methods: CQT and MFCC.

a) CQT:

CQT is a time-frequency analysis approach used to extract features. The CQT, in contrast to the more common Fourier Transform, separates the signal into logarithmically spaced frequency bins. This means that the bins are positioned so that each octave has an equal number of bins. This is useful because it replicates the human auditory system's logarithmic perception of frequency.

The CQT, like the Fourier Transform, translates a signal from time to frequency. The CQT, on the other hand, gives a frequency representation that is more closely aligned with human perception due to its logarithmic frequency bin spacing. CQT is very useful in audio signal processing applications such as music analysis, audio synthesis, and feature extraction for tasks like audio classification and identification. It provides a frequency representation that better captures the tonal characteristics of audio signals, making it well-suited for applications that require such features. The CQT feature extraction method is shown in Figure 2.

$$C_i^n = CQT(S_r^n) \tag{2}$$

$$Features_{CQT}^n = \begin{bmatrix} C_1^1 & C_2^1 & \cdots & C_i^1 \\ C_1^2 & C_2^2 & \cdots & C_i^2 \\ & \vdots & \vdots & \vdots \\ C_1^n & C_2^n & \cdots & C_i^n \end{bmatrix} \tag{3}$$

Where $Features_{CQT}^n$ are features of CQT from different ASR models and samples.

In equation (2) the audio samples are given to the CQT feature extraction method to get acoustic characteristics. They are shown in equation (3).

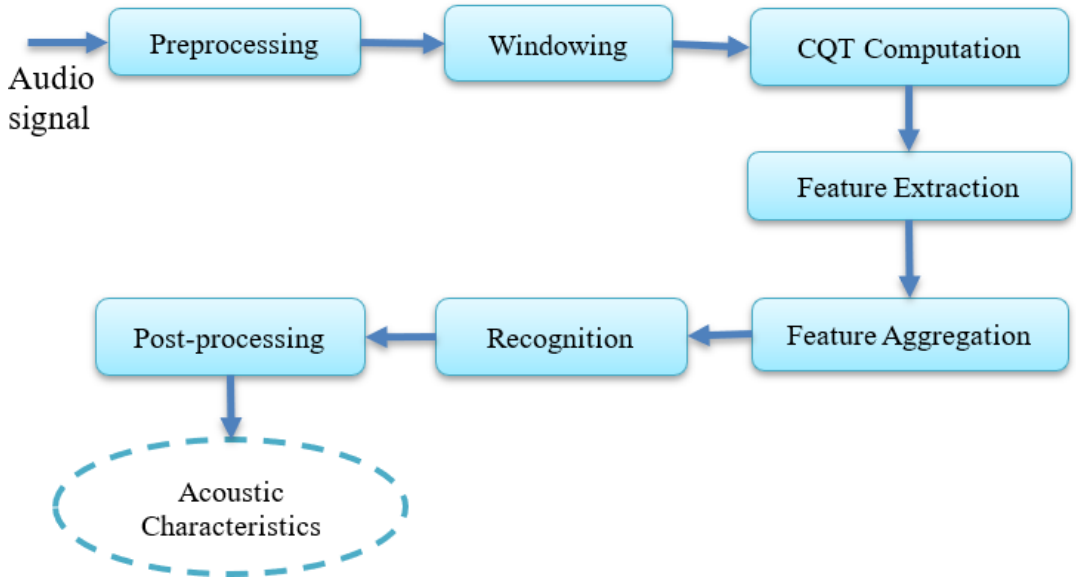


Fig 2: CQT Feature Extraction method

b) Mel - Frequency Cepstral Coefficient (MFCC):

The most popular feature extraction method in speech and audio processing is called MFCC. The spectrum qualities of sound are described by MFCCs in a form that is suitable for a range of machine learning applications, such as music analysis and speech recognition [19]. A group of

coefficients that represent the sound source's power spectrum form is known as an MFCC. The Mel - Scale is used to simulate how the human ear hears sound frequency after the raw audio input has been transformed into a frequency domain using a technique similar to the Discrete Fourier Transform (DFT).

Lastly, the mel-scaled spectrum is used to compute the cepstral coefficients. Because they eliminate unnecessary information while highlighting audio signal elements essential to human speech perception, MFCCs are very advantageous. They can therefore be used for tasks including speech-to-text conversion, emotion detection, and speaker recognition. The MFCC feature extraction method is shown in Figure 3.

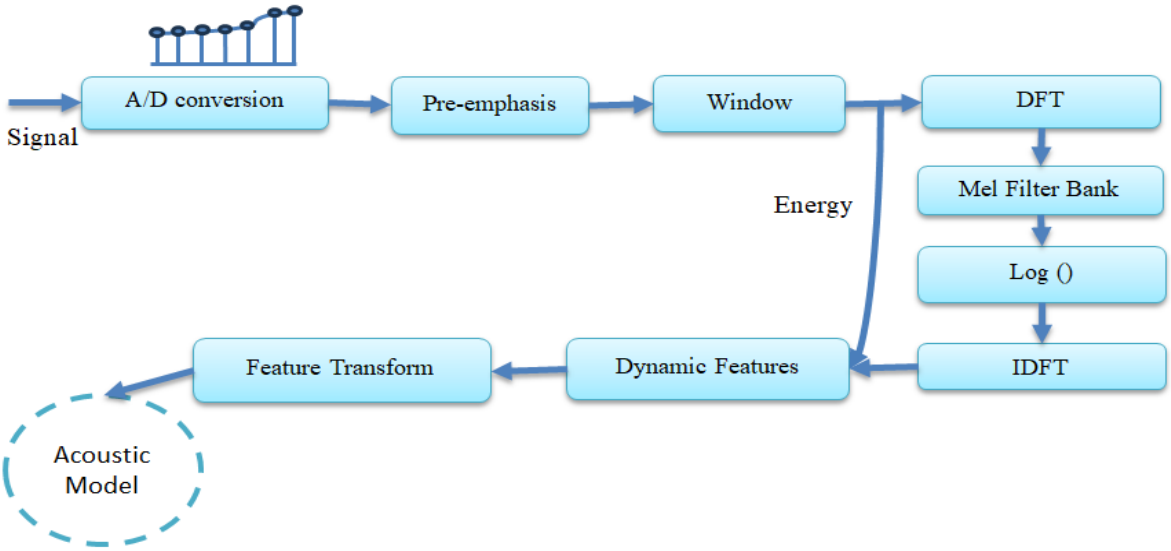


Fig 3: MFCC feature extraction method

$$M_i^n = MFCC(S_i^n) \tag{4}$$

$$Features_{MFCC}^n = \begin{bmatrix} M_1^1 & M_2^1 & \cdots & M_i^1 \\ M_1^2 & M_2^2 & \cdots & M_i^2 \\ & \vdots & \vdots & \vdots \\ M_1^n & M_2^n & \cdots & M_i^n \end{bmatrix} \tag{5}$$

Where $Features_{MFCC}^n$ is MFCC features for differentASR devices and samples. Calculate threshold value which is calculated by comparing the same device sample features

3) Model Training:

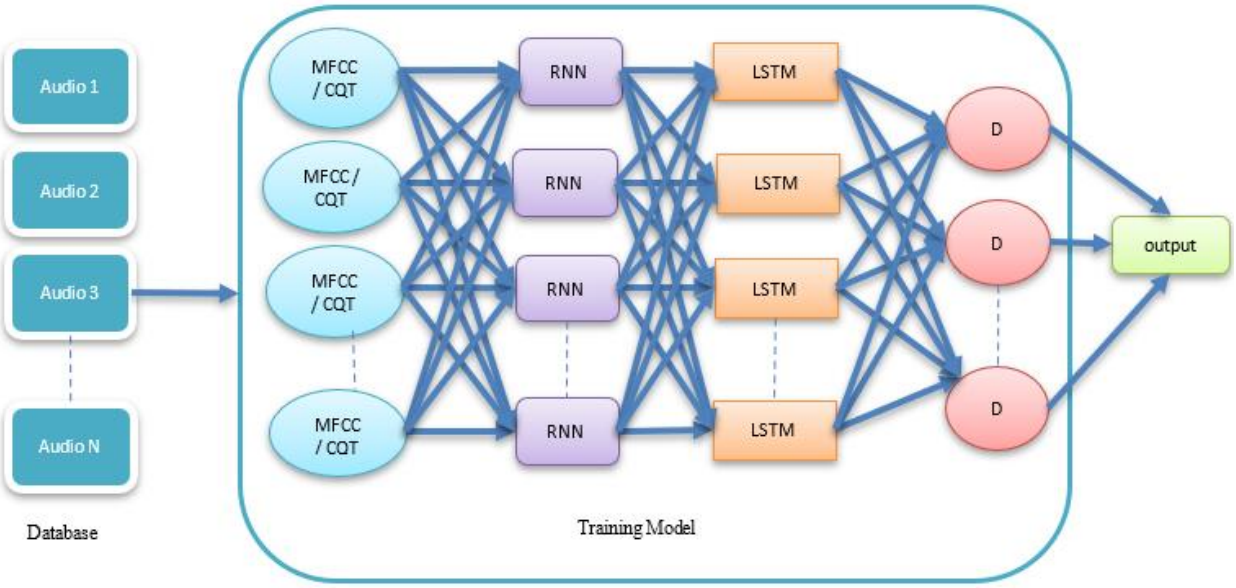


Fig 4: Model Training

In speech and audio processing, characteristics are extracted using MFCC or CQT. It highlights human auditory awareness by capturing the spectrum characteristics of sound. By effectively encoding audio data, MFCC enables machine learning algorithms can categorize, identify, and examine auditory patterns. They are extensively utilized in audio recognition, speech identification, and the analysis of music.

a) RNN:

A deep learning model designed to analyze and convert sequential data inputs into specific sequential data outputs is called an RNN. RNNs are composed of neurons, which are nodes for processing data that work together to do complex tasks. There are three levels of neurons: input, output, and hidden shown in Fig.5. While the output layer generates the final product, the input layer receives the data to process. Prediction, analysis, and data processing happen at the hidden layer.

The information is moving in a single direction through the feed forward neural network (FNN): through hidden layers, from the input layer to the output layer. There is a direct flow of information throughout the network. The prediction of the future is not good for FNNs because they can't remember what has been communicated to them. Since the FNN is only looking at existing inputs, it does not have a deadline. Apart from its instruction, it is incapable of remembering anything from the past. Information is looped across an RNN. It takes into consideration the information it has acquired from prior inputs in addition to that of current input when deciding on a decision.

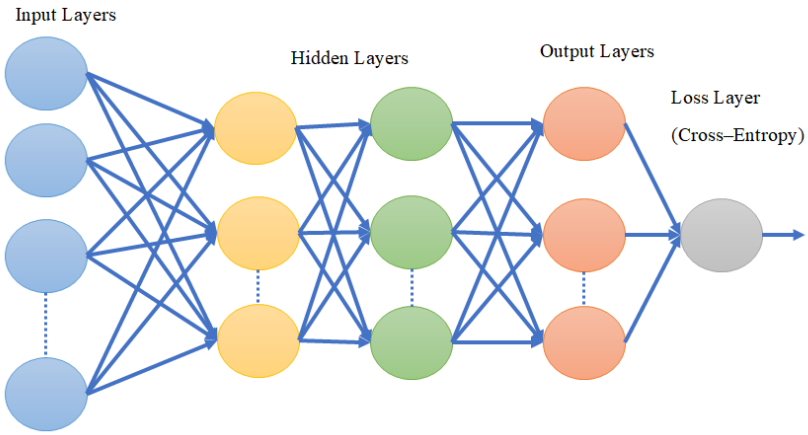


Fig 5: RNN Architecture

b) LSTM:

RNNs are extended by LSTM networks. The RNN's layers are constructed from LSTMs. Three gates are present in a long short-term memory cell: input, forget, and output. These gates control whether to allow data to enter the system (input gate), discard data that isn't needed (forget gate), or allow data to affect the output at the current time step (output gate).

$$B_m^{CQT} = RNN(LSTM(Features_{CQT}^n)) \text{ using CQT} \quad (6)$$

$$B_m^{MFCC} = RNN(LSTM(Features_{MFCC}^n)) \text{ using MFCC} \quad (7)$$

4) Model Evaluation:

Assess the trained model's performance on the test set by using suitable evaluation metrics, like F1-score, accuracy, precision, and recall. Analyze any misclassifications to identify areas for improvement. Higher values for precision, recall, and F1 score indicate better performance, with the best score possible shown in Figure 6 for CQT and Figure 8 for MFCC.

5) Model Testing:

The new audio sample is given to the trained model then it predicts the output. It recognizes the ASR device.

For CQT

$$PD_{CQT} = B_m^{CQT}(a_t) \quad (8)$$

$$where\ a_t\ is\ test\ sample$$

$$Features_{a_t}^{CQT} = \begin{bmatrix} C_{t_1}^{1} & C_{t_2}^{1} & \cdots & C_{t_i}^{1} \\ C_{t_1}^{2} & C_{t_2}^{2} & \cdots & C_{t_i}^{2} \\ \vdots & \vdots & \vdots & \\ C_{t_1}^{n} & C_{t_2}^{n} & \cdots & C_{t_i}^{n} \end{bmatrix} \text{using CQT} \quad (9)$$

$$if\ Features_{a_t}^{CQT} \geq Th_{CQT} : \text{Predicts the model} \quad (10)$$

Where Th_{CQT} is a threshold value for CQT

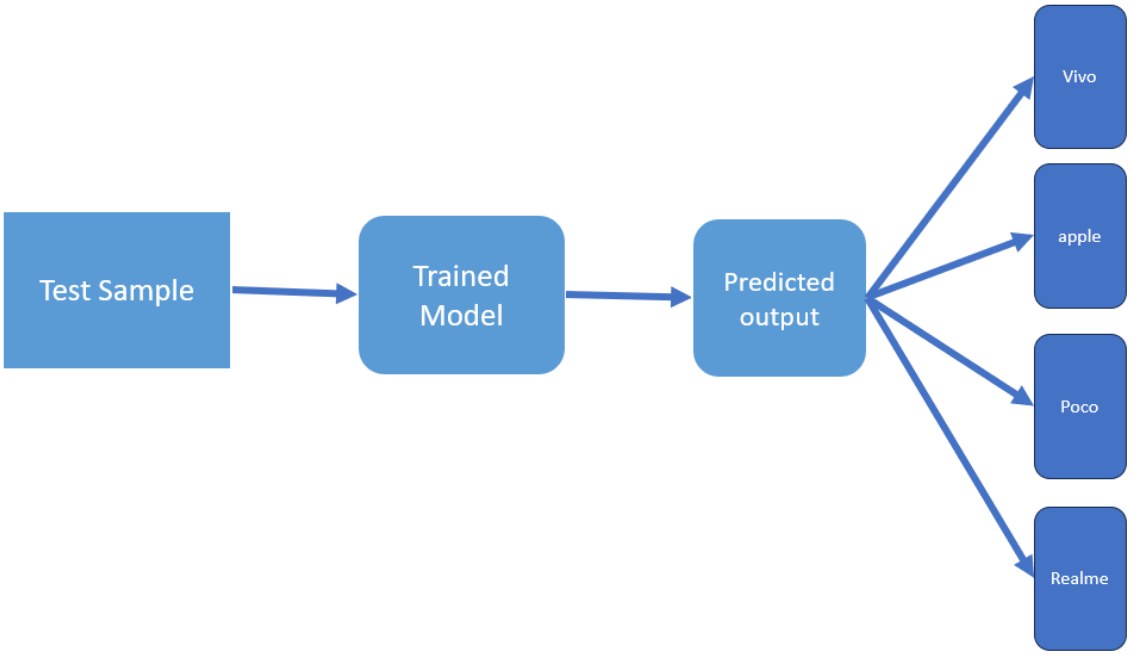


Fig 6: Model testing

For MFCC:

MFCC model testing is shown in the following equations.

$$PD_{MFCC} = B_m^{MFCC}(a_t) \tag{11}$$

$$where\ a_t\ is\ test\ sample$$

$$Features_{a_t}^{MFCC} = \begin{bmatrix} M_{t_1}^{1} & M_{t_2}^{1} & \cdots & M_{t_i}^{1} \\ M_{t_1}^{2} & M_{t_2}^{2} & \cdots & M_{t_i}^{2} \\ \vdots & \vdots & \vdots & \\ M_{t_1}^{n} & M_{t_2}^{n} & \cdots & M_{t_i}^{n} \end{bmatrix} \tag{12}$$

$$if\ Features_{a_t}^{MFCC} \geq Th_{MFCC} \text{ :Predicts the model} \tag{13}$$

Where Th_{MFCC} is a threshold value for MFCC

6) Deployment:

Once the model achieves satisfactory performance, deploy it for real-world use. It can now classify the source device of new audio recordings based on their features.

The suggested approach advances audio forensic analysis significantly by exploiting the benefits of MFCC and temporal modeling techniques, promising increased efficacy in a variety of forensic contexts.

Overall, findings highlight the need to use proper feature extraction approaches in ASR device detection tasks. MFCC's exceptional performance demonstrates its potential for reliably identifying source devices based on their distinct audio characteristics, paving the door for more reliable and effective forensic analysis in digital investigations.

IV. Experimental Results

The primary goal of this study is to establish the efficacy of the proposed approach using extensive experimental assessments. To demonstrate the durability and applicability of methodology across numerous forensic scenarios by leveraging diverse datasets that include recordings from a variety of source devices. Through extensive testing and validation, to give empirical evidence of the model's improved performance in contrast to other methodologies.

The performance of the proposed ASR-CQT-RNN and ASR-MFCC-RNN methods are evaluated on four ASR device datasets. The dataset collection process includes the following steps. Audio samples are downloaded using ITU-T test signals from telecommunications systems. It takes around 12 minutes to merge audio samples into large corpus data. The recorded audio samples using different devices, where each device Split 12-minute audio into 70 samples from the given data in an efficient way. The sequence in which the samples were combined from TIMIT was such that the long speech data was separated into 560 short sample segments, each lasting around 10 seconds, based on the quantity and duration of the TIMIT corpus. The ASR devices used in this work are iPhone, Realme, Vivo, and Poco. This also emphasizes the significance of taking into account a variety of aspects, such as recording settings, background noise, and device characteristics, while developing experiments and preprocessing procedures for audio source recognition.

The performance metrics for ASR-CQT-RNN and ASR-MFCC-RNN methods are shown in Table 1 and Table 2. Furthermore, confusion matrix for both approaches are supplied to help visualize their performance in identifying audio sources shown in Figures 7 & 8. Table 1 and Table 2 reported accuracy rates of 79.94% for ASR-CQT-RNN and 96.49% for ASR-MFCC-RNN showing that ASR-MFCC-RNN surpasses ASR-CQT-RNN in this situation. The testing results showed that ASR-MFCC-RNN regularly surpassed ASR-CQT-RNN in terms of accuracy, confirming its superiority as a feature extraction method for recording device recognition.

Table 1: Performance Metrics for ASR-CQT-RNN

	Precision	Recall	f1 – score	support
Vivo	0.86	1.00	0.92	12
Poco	0.72	0.87	0.79	15
Apple	1.00	0.47	0.64	15
Realme	0.72	0.87	0.79	15
Accuracy	-	-	0.79	57
Macro avg	0.83	0.80	0.78	57
Weighted avg	0.82	0.79	0.78	57

Figure 7 illustrates a confusion matrix for the ASR-CQT-RNN. It is a table that shows how well a classification model performs when applied to a set of test data whose actual values are known. Here are the data from the matrix:

The rows correspond to the real classes (True labels): Vivo, Poco, Apple, and Realme. The columns indicate the model's projected classes (projected labels), which are Vivo, Poco, Apple, and Realme. The matrix displays the number of correct and incorrect predictions made by the model, with the diagonal indicating correct predictions. These are the specific counts:

In Figure 7, Vivo had 12 correct guesses; 0 incorrect, Poco had 13 correct predictions, 2 wrong (both predicted Realme), and Apple had seven right predictions, 1 wrong prediction for Vivo, three for Realme, and four for Poco. Realme had 13 right guesses, one bad prediction as Vivo, and one as Poco.

For example, Vivo and Poco both exhibit a high number of correct predictions (dark blue), with 12 and 13 respectively, whereas Apple has a more spread-out confusion, with 7 correct but multiple misclassifications among other brands.

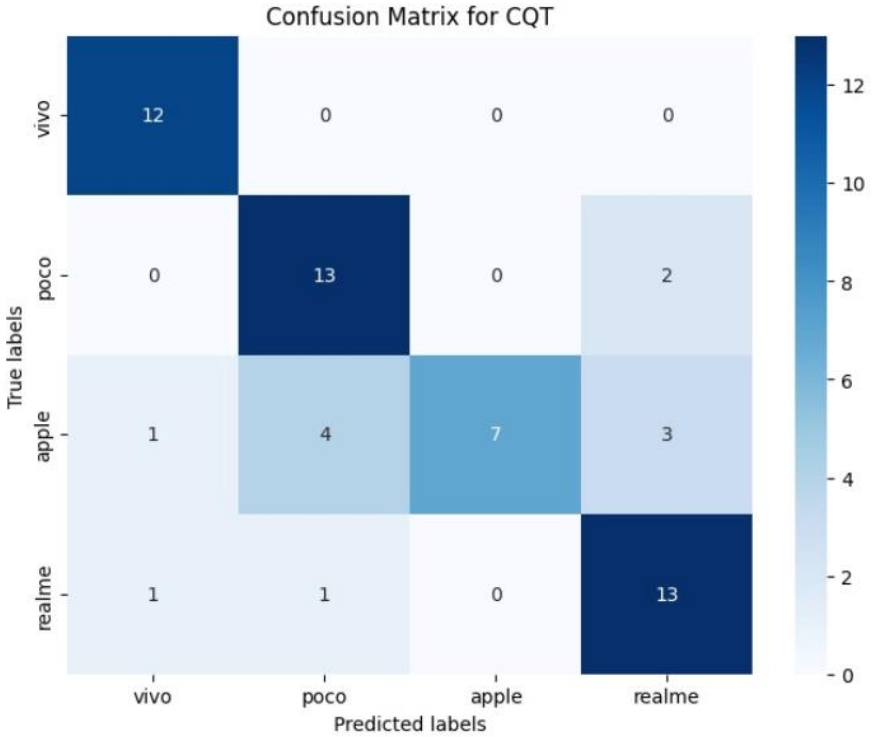


Fig 7: Confusion Matrix ASR-CQT-RNN

Table 2: Performance metrics for ASR-MFCC-RNN

	Precision	recall	F1 - score	support
Vivo	1.00	1.00	1.00	12
Poco	0.88	1.00	0.94	15
Apple	1.00	0.87	0.93	15
Realme	1.00	1.00	1.00	15
accuracy			0.96	57
Macro avg	0.97	0.97	0.97	57
Weighted avg	0.97	0.96	0.96	57

Figure 8 shows confusion matrix of ASR-MFCC-RNN, in that dark navy blue color represents true positives and the light color represents true negatives. False predictions are not possible. For Vivo 12 correct guesses and 0 incorrect, Poco had 15 correct predictions, Apple had 13 right predictions and two wrong predictions as Poco, and Realme had 15 right predictions.

CQT, while commonly used in audio analysis, did not consistently perform well in tests. On the other hand, MFCC evolved as a more reliable and accurate feature representation method for recognizing audio source recording devices. MFCC, which is derived from the human auditory

system's frequency resolution properties, collects critical information about the spectrum envelope of audio sources. This format efficiently captures the distinguishing properties of various recording devices, allowing for accurate device recognition. Furthermore, combining MFCC with a structured representation learning model, which includes RNNs and LSTM, improves the accuracy of device identification.

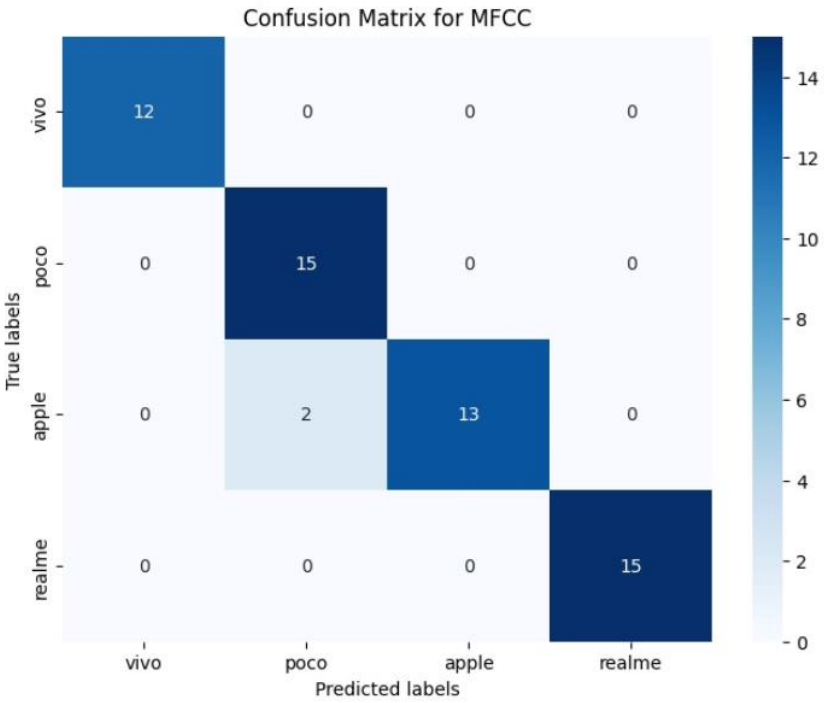


Fig 8: Confusion Matrix of ASR-MFCC-RNN

Fig 9: Accuracy Plot

Training and validation accuracy over epochs for MFCC and CQT feature extraction strategies are highlighted in Figure 9. The number of training epochs, or iterations, through the dataset, is

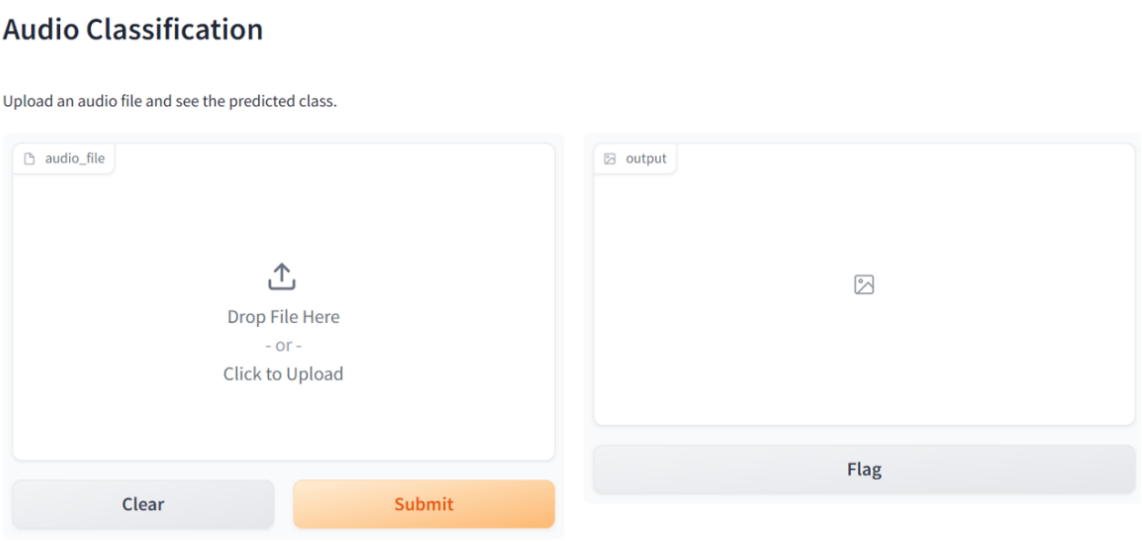
represented by the X-axis. Every epoch represents a single pass over the whole training set. The accuracy of the model is shown on the Y-axis for both the training and validation datasets. The percentage of correctly identified examples relative to all instances is commonly used to quantify accuracy. The training accuracy curve illustrates how the model's accuracy varies over epochs on the training dataset. The validation accuracy curve illustrates how the model's accuracy varies over epochs on the validation dataset. The intersection of the training and validation curves can have important implications. A model may be overfitting if its validation accuracy is regularly lower than its training accuracy. This indicates that the model is fitting the training data too closely and is not generalizing well to new data. Good generalization is shown if the validation accuracy is similar to the training accuracy. The comparison of accuracy rates, the presentation of confusion matrices, and the evaluation of various ASR devices and potential problems all help to provide a full understanding of the experimental results.

Table 3: Comparison of the proposed work with stat-of-art-works

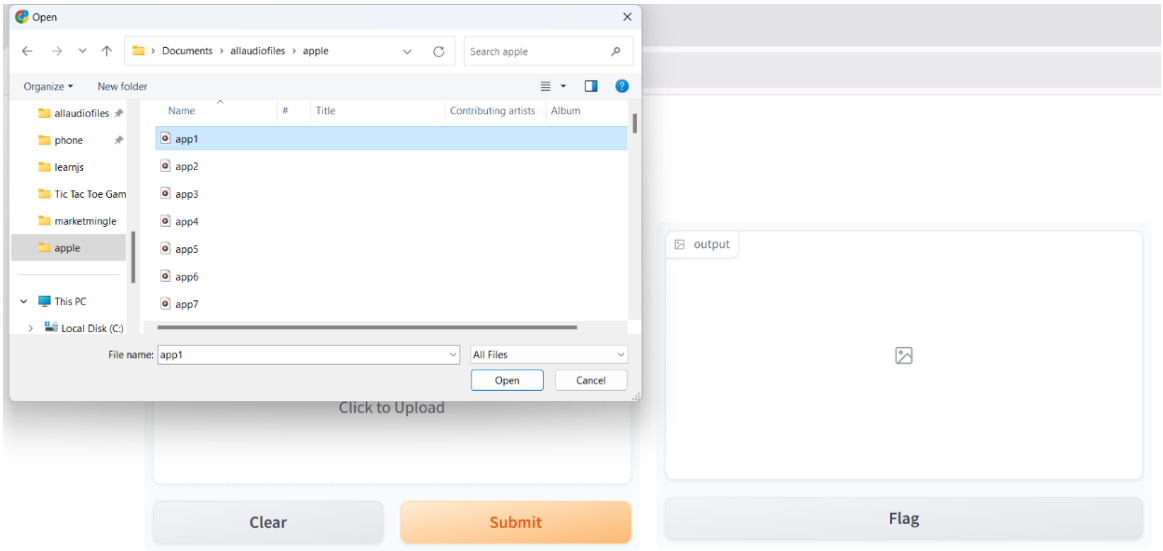
Previous Papers	**Accuracy**
Yuechi Jiang et.al (2019)[14] *Scheme-1 (SRC)*	88.37%
Yuechi Jiang et.al (2019)[14] *Scheme-2 (SVM)*	89.04%
Jungbeom Ko1 et.al [16]	91.41%
ASR-CQT-RNN (Proposed method-1)	79.94%
ASR-MFCC-RNN (Proposed method-2)	96.49%

The performance of the proposed work in terms of accuracy is compared with state-of-art works which is shown in Table 3. It is observed that the accuracy of this model is more when compared to previous works. This suggests that the combination of MFCC features with RNN and LSTM models was highly effective for the task.

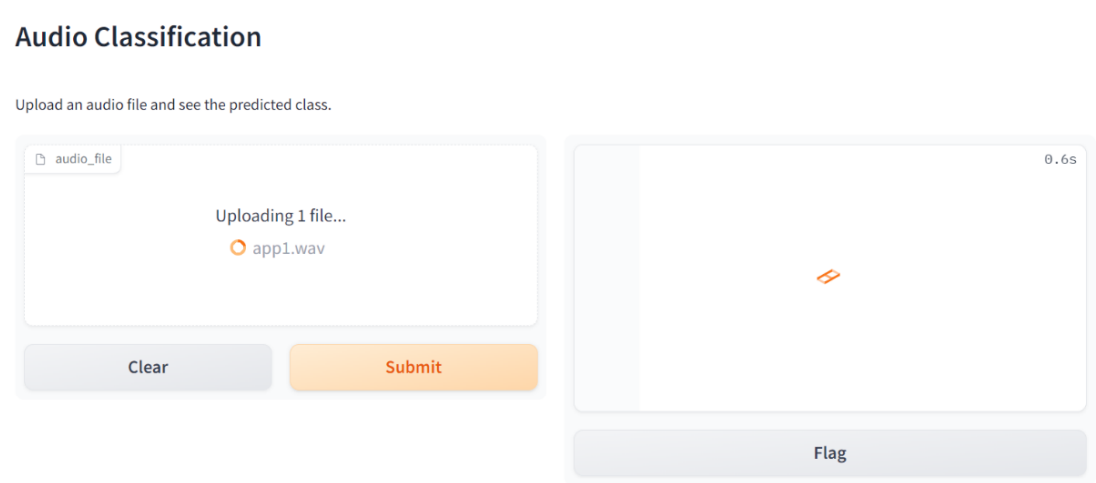
This proposed work is also implemented for the user interface which is shown in Figure 10. The user interface screen is divided into two parts, one part of the screen is to upload the test audio signal and the second part of the screen is to get output ASR device shown in Figure 10(a) and (b). Whenever the test audio sample is uploaded which is shown in Figure (c) and Figure (d), then the proposed code will be run in the background and it will produce the predicted source device which is shown in Figure (e).


Audio Classification
Upload an audio file and see the predicted class.
audio_file
Drop File Here
- or -
Click to Upload
Clear
Submit
output
Flag

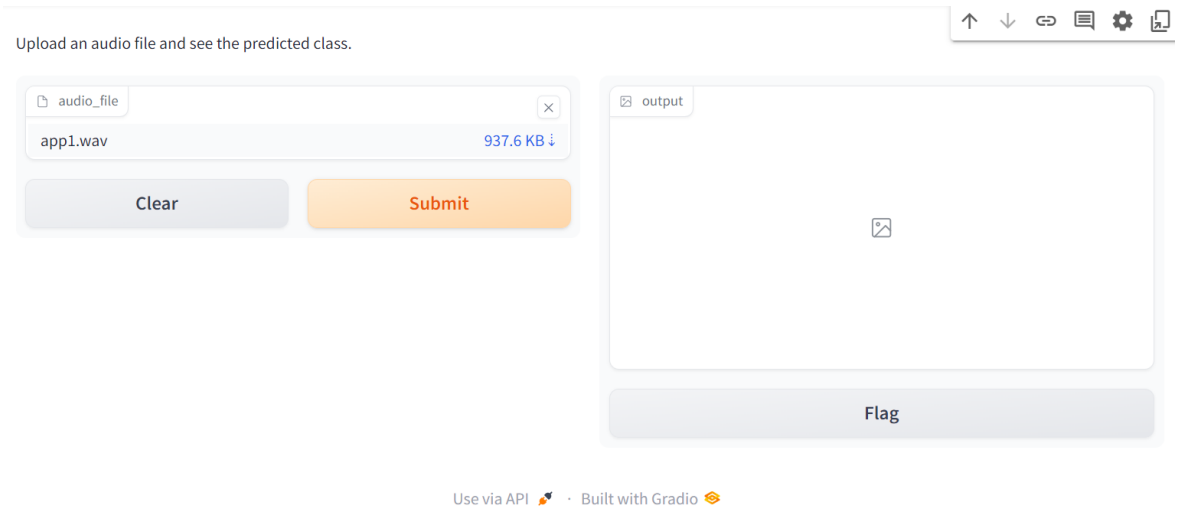
(a)


Open
Documents › allaudiofiles › apple
Search apple
Organize
New folder
allaudiofiles
phone
learnjs
Tic Tac Toe Gam
marketmingle
apple
This PC
Local Disk (C:)
Name
#
Title
Contributing artists
Album
app1
app2
app3
app4
app5
app6
app7
File name: app1
All Files
Open
Cancel
Click to Upload
Clear
Submit
output
Flag

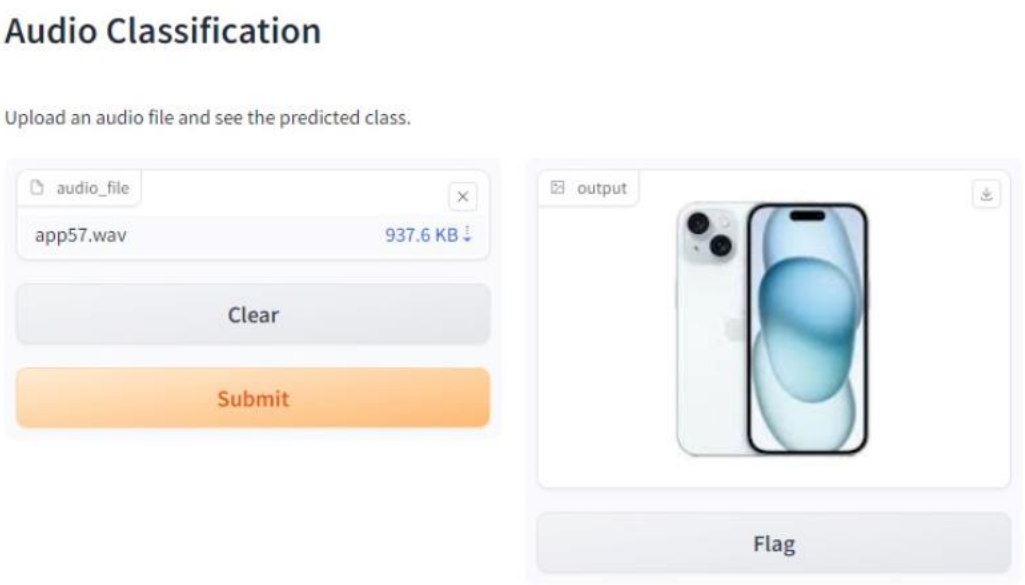
(b)


Audio Classification
Upload an audio file and see the predicted class.
audio_file
Uploading 1 file...
app1.wav
Clear
Submit
0.6s
Flag

(c)



(d)



(e)

Fig 10: User Interface

V. Conclusion & Future Scope

In conclusion, the findings show that integrating MFCC with RNN and LSTM improves ASR device recognition accuracy. While MFCC and CQT were compared, it was discovered that CQT did not consistently perform well on this task. The performance of this proposed work in terms of accuracy is compared with state-of-art works and using MFCC with RNN and LSTM resulted in the 96.49% accurate identification of source devices based on distinctive audio characteristics. This work is implemented in frontend design for a better user interface to identify the ASR device of a given test audio signal.

Furthermore, intend to improve the resilience of this model by including approaches for dealing with fluctuations in the audio database, such as ambient noise and recording settings. Future research will focus on broadening the applicability of this approach to solve growing audio forensics difficulties, such as detecting deepfake audio recordings and identifying tampered or modified content.

References

[1] Z. Wang, J. Zhan, G. Zhang, D. Ouyang, and H. Guo, "An End-to-End Transfer Learning Framework of Source Recording Device Identification for Audio Sustainable Security," *Sustain.*, vol. 15, no. 14, pp. 1–22, 2023, doi: 10.3390/su151411272.

[2] C. Zeng, S. Kong, Z. Wang, K. Li, and Y. Zhao, "Digital Audio Tampering Detection Based on Deep Temporal–Spatial Features of Electrical Network Frequency," *Inf.*, vol. 14, no. 5, pp. 1–22, 2023, doi: 10.3390/info14050253.

[3] R. Buchholz, C. Kraetzer, and J. Dittmann, "Microphone classification using fourier coefficients," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5806 LNCS, pp. 235–246, 2009, doi: 10.1007/978-3-642-04431-1_17.

[4] H. Malik and J. W. Miller, "Microphone identification using higher-order statistics," in *Proceedings of the AES International Conference*, 2012, pp. 134–143.

[5] C. Hanilçi and T. Kinnunen, "Source cell-phone recognition from recorded speech using non-speech segments," *Digit. Signal Process.*, vol. 35, pp. 75–85, 2014, doi: 10.1016/j.dsp.2014.08.008.

[6] Ö. Eskidere, "Source microphone identification from speech recordings based on a Gaussian mixture model," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 22, no. 3, pp. 754–767, 2014, doi: 10.3906/elk-1207-74.

[7] K. J. Piczak, "Environmental Sound Classification with Convolutional Neural Networks," in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP), Boston, USA*, 2015, pp. 1–6.

[8] J. Salamon and J. P. Bello, "Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification," *IEEE Signal Process. Lett.*, vol. 24, no. 3, pp. 279–283, 2017, doi: 10.1109/LSP.2017.2657381.

[9] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep Clustering: Discriminative Embeddings for Segmentation and Separation," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2016, pp. 31–35.

[10] J. F. Gemmeke *et al.*, "Audio Set: An ontology and human-labeled dataset for audio events," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, pp. 776–780, 2017, doi: 10.1109/ICASSP.2017.7952261.

[11] S. Qi, Z. Huang, Y. Li, and S. Shi, "Audio recording device identification based on deep learning," in *2016 IEEE International Conference on Signal and Image Processing, ICSIP 2016*, 2017, pp. 426–431, doi: 10.1109/SIPROCESS.2016.7888298.

[12] G. Baldini and I. Amerini, "Smartphones identification through the built-in microphones with convolutional neural network," *IEEE Access*, vol. 7, pp. 158685–158696, 2019, doi: 10.1109/ACCESS.2019.2950859.

[13] Q. Yan, R. Yang, and J. Huang, "Robust copy-move detection of speech recording using similarities of pitch and formant," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 9, pp. 2331–2341, 2019, doi: 10.1109/TIFS.2019.2895965.

[14] Y. Jiang and F. H. F. Leung, "Smartphones identification through the built-in microphones with convolutional neural network," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 11, pp. 2875–

2886, 2019, doi: 10.1109/TIFS.2019.2911175.

[15] D. Renza, J. Vargas, and D. M. Ballesteros, “Robust speech hashing for digital audio forensics,” *Appl. Sci.*, vol. 10, no. 1, 2020, doi: 10.3390/app10010249.

[16] J. Ko, H. Kim, and J. Kim, “Real-Time Sound Source Localization for Low-Power IoT Devices Based on Multi-Stream CNN,” *Sensors*, vol. 22, no. 12, 2022, doi: 10.3390/s22124650.

[17] M. A. Qamhan, H. Altaheri, A. H. Meftah, G. Muhammad, and Y. A. Alotaibi, “Digital Audio Forensics: Microphone and Environment Classification Using Deep Learning,” *IEEE Access*, vol. 9, pp. 62719–62733, 2021, doi: 10.1109/ACCESS.2021.3073786.

[18] A. Giganti, L. Cuccovillo, P. Bestagini, P. Aichroth, and S. Tubaro, “Speaker-Independent Microphone Identification in Noisy Conditions,” 2022, [Online]. Available: http://arxiv.org/abs/2206.11640.

[19] M. K. Singh, “Multimedia application for forensic automatic speaker recognition from disguised voices using MFCC feature extraction and classification techniques,” *Multimed. Tools Appl.*, no. 0123456789, 2024, doi: 10.1007/s11042-024-18602-4.