iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79313

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Speech Emotion Recognition Using Machine Learning Approach

Priyanshu Singh[1], Athrave Singh Parihar[2], Vedant Kumar Chaubey[3], Tushar Jain[4], Abha Choubey[5]
[1,2,3,4]*Student, Computer Science and Engineering, Shri Shankaracharya Technical Campus*
[5]*Assisant Professor, Computer Science and Engineering, Shri Shankaracharya Technical Campus*

Abstract: ***Speech Emotion Recognition (SER) has emerged as a critical field in artificial intelligence, enabling systems to interpret human emotions through speech signals. This research proposes a comprehensive SER system utilizing Mel-Frequency Cepstral Coefficients (MFCCs) as primary features extracted from audio signals. The system is trained and evaluated on a combined dataset consisting of RAVDESS and TESS, comprising 5,252 labeled audio samples across eight emotional categories. A comparative study is conducted between a traditional Decision Tree classifier and a deep learning-based One-Dimensional Convolutional Neural Network (1D CNN). The Decision Tree model achieves an accuracy of approximately 68%, whereas the CNN achieves around 85.5% validation accuracy, demonstrating superior performance. In addition to model development, a desktop-based application using CustomTkinter is implemented for real-time emotion detection from microphone input or audio files. The research highlights the importance of feature extraction, model selection, and deployment considerations while presenting a scalable and practical solution for SER.***

Keywords: Speech Emotion Recognition, MFCC, CNN, Decision Tree, Deep Learning, TensorFlow, RAVDESS, TESS.

I. INTRODUCTION

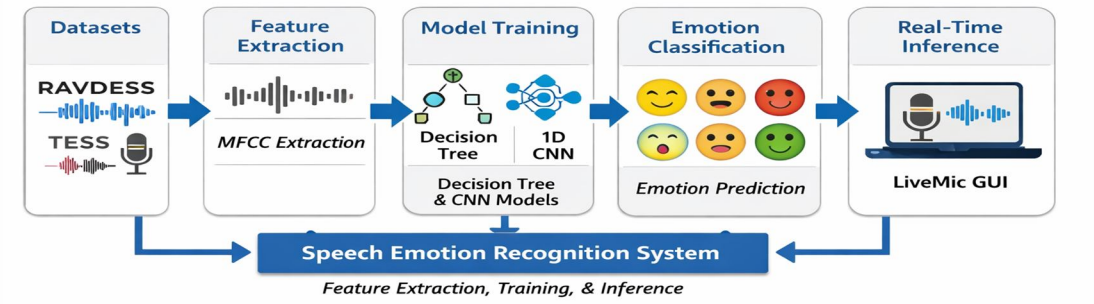
Speech is one of the most expressive forms of human communication, conveying not only linguistic information but also emotional context. Understanding emotions from speech is essential in numerous applications such as virtual assistants, healthcare monitoring, education systems, and customer service platforms. However, accurately identifying emotions from speech is a challenging task due to various factors such as speaker variability, background noise, recording conditions, and subjective interpretation of emotions.

Traditional approaches for speech emotion recognition rely heavily on handcrafted features and classical machine learning algorithms. While these methods provide some level of accuracy, they often fail to generalize well across different datasets and real-world scenarios. With the advancement of deep learning, more robust and scalable solutions have been developed that can automatically learn features from raw or processed audio signals.

This project aims to develop an effective speech emotion recognition system by leveraging MFCC features and comparing the performance of Decision Tree and Convolutional Neural Network models. The integration of a real-time desktop application further enhances the practical usability of the system.

The main contributions of this research include:

1) Implementation of a complete SER pipeline
2) Comparative analysis of traditional and deep learning models
3) Development of a real-time desktop application
4) Identification of practical challenges and limitations



[Figure: SER Workflow / MFCC Visualization Here]

II. LITERATURE REVIEW

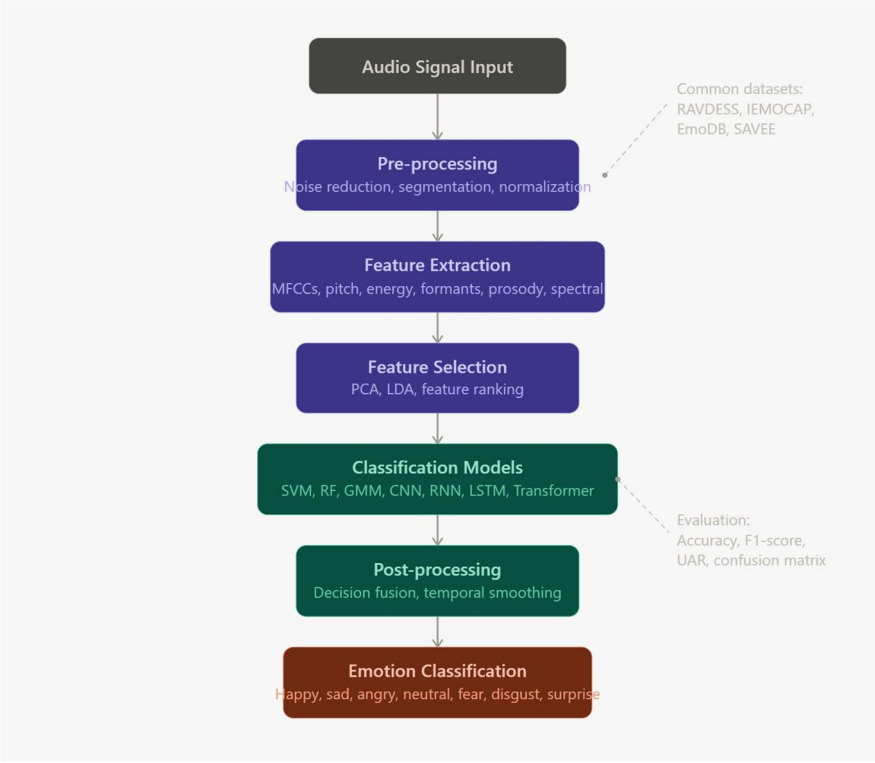
Speech Emotion Recognition (SER) focuses on identifying emotional states from speech signals using acoustic features and classification models. A widely used approach involves extracting Mel-Frequency Cepstral Coefficients (MFCCs) or Mel spectrograms using tools like *Librosa*, which provide compact and perceptually relevant representations of audio signals.

For classification, traditional machine learning models such as Decision Trees, Support Vector Machines (SVM), and Random Forests are commonly used due to their simplicity and interpretability. However, these models operate on fixed-length feature vectors and often fail to capture temporal dependencies in speech. To address this, deep learning frameworks like *TensorFlow/Keras* enable models such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), which can learn temporal and hierarchical patterns directly from audio features, leading to improved performance.

Datasets such as RAVDESS and TESS are widely used benchmarks in SER research. However, they have limitations including acted (non-spontaneous) speech, limited speaker diversity, and domain differences when combined. Additionally, random train-test splits may include the same speakers in both sets, leading to inflated accuracy compared to more robust evaluation methods like speaker-independent testing.

This project follows a similar pipeline using 40 MFCC features with mean pooling, stored via Joblib, and applies both a Decision Tree classifier and a Conv1D model on the combined RAVDESS+TESS dataset (~5,200 samples, eight classes). A desktop-based inference system using CustomTkinter, sounddevice, and Matplotlib enables real-time prediction from microphone or WAV input.

The contribution of this work lies in providing an integrated pipeline with a fair comparison between classical and deep learning models, rather than claiming state-of-the-art performance.



[Figure : Data Flow Diagram (DFD)]

Domain Stack vs This Project

LAYER	COMMON PRACTICE	THIS PROJECT
FEATURES	MFCC / Spectrogram	40 MFCC (mean pooled)
MODELS	SVM, RF, CNN, RNN	Decision Tree + Conv1D
DATA	RAVDESS, TESS	Combined dataset
DEPLOYMENT	Notebooks / APIs	Desktop GUI

III. TECHNOLOGY TABLE

Technology	Description	Usage
MFCC	Mel-frequency cepstral coefficients	Feature extraction
Librosa	Audio processing library	Audio loading and MFCC extraction
TensorFlow/Keras	Deep learning framework	CNN implementation
Scikit-learn	Machine learning library	Decision Tree model
CustomTkinter	GUI library	Desktop application
Matplotlib	Visualization library	Graph plotting
NumPy	Numerical computing	Data processing
Joblib	Serialization library	Saving features

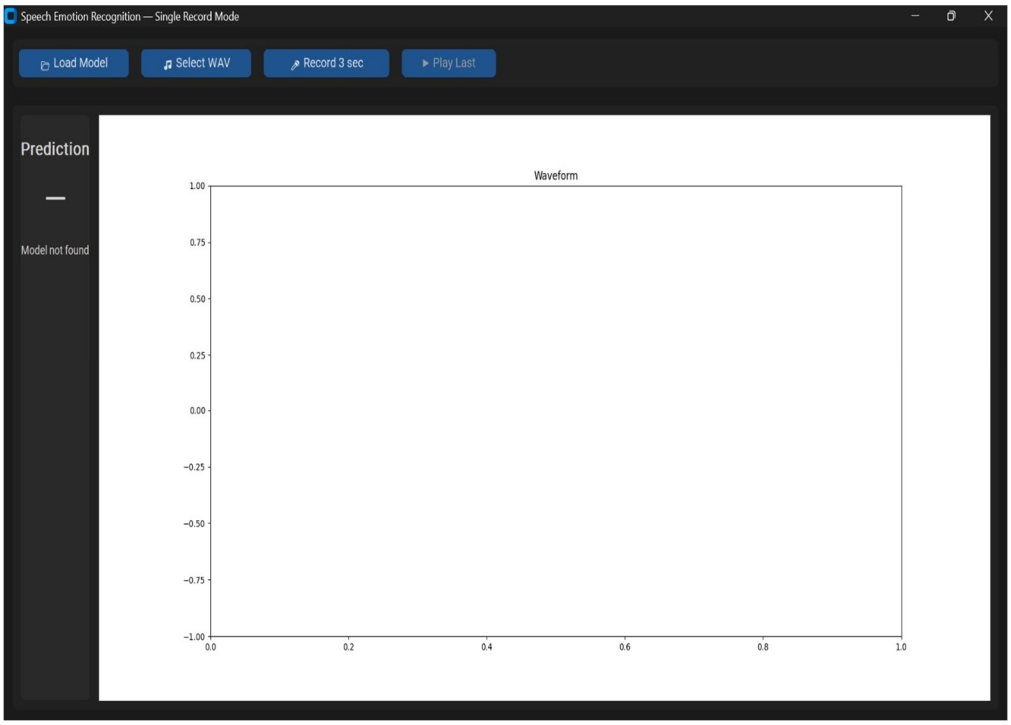
IV. METHODOLOGY

This section describes the system architecture, workflow, and implementation of the Speech Emotion Recognition (SER) system. The project follows a modular design consisting of feature extraction, model training, and real-time inference. Unlike web-based systems, this implementation uses a file-based approach with Joblib and Keras .h5 models instead of databases or APIs.

A. System Architecture

The system is divided into four main components:

Component	*Description*
Feature Extraction	Extracts 40 MFCC features from audio using Librosa
Model Training	Implements Decision Tree and Conv1D models
Feature Storage	Stores features using Joblib (X.joblib, y.joblib)
Inference System	Desktop GUI for real-time prediction



[Figure : System Architecture Diagram]

B. Layered Structure

The system follows a logical layered architecture:

- Presentation Layer: Desktop GUI (LiveMic.py) built using CustomTkinter for user interaction.
- Application Layer: Controls workflow such as audio recording, file selection, and prediction.
- Processing Layer: Handles audio loading, MFCC extraction, and model prediction using Librosa and TensorFlow/Keras.
- Persistence Layer: Stores features (X.joblib, y.joblib) and trained model (.h5) in the local filesystem.

C. Workflow of the System

1) Training Workflow

- Audio files are collected from RAVDESS and TESS datasets
- MFCC features (40 coefficients) are extracted and stored
- Data is split using train_test_split
- Decision Tree and CNN models are trained
- Trained CNN model is saved as .h5

2) Inference Workflow

- User records audio (3 seconds) or selects a WAV file
- Audio is processed and MFCC features are extracted
- Features are reshaped to (1, 40, 1)
- Model predicts emotion using softmax
- Output is displayed in GUI

D. Data Flow Explanation

The system processes data in three main stages:

Feature Generation

Raw Audio → Librosa Load → MFCC Extraction → Mean Pooling → Feature Vector (40) → Stored via Joblib

Training Flow

Feature Data → Train-Test Split → Model Training → Evaluation

Inference Flow

Audio Input → MFCC → Reshape → Model Prediction → Emotion Output

E. Implementation Details

1) Feature Extraction: Audio is loaded using Librosa and converted into 40 MFCC features. Mean pooling is applied to create fixed-length vectors.

2) Model Training: A Decision Tree is used as a baseline. A Conv1D model is trained using TensorFlow/Keras with softmax output for 8 emotion classes.

3) Inference System: The GUI application records audio or loads files, extracts features, and predicts emotions in real time.

V. RESULTS AND DISCUSSION

A. Achievements

The system performs eight-class speech emotion classification using 40 mean-pooled MFCC features on a combined RAVDESS + TESS dataset (~5,200 samples). Using train_test_split (33% test, random_state = 42), the dataset is divided into:

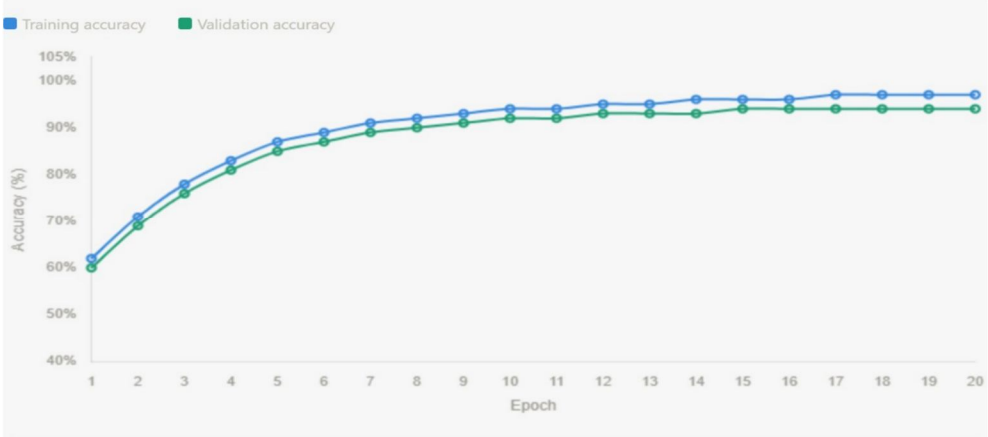
- Training Samples: 3,518
- Testing Samples: 1,734

Two models were evaluated:

- Decision Tree (Baseline):
  Achieved approximately 68% accuracy with a macro F1-score of ~0.67. Performance varies across classes (F1 ≈ 0.57–0.80), indicating limited ability to distinguish complex emotions.

- 1-D CNN (Deep Learning Model):
  After 200 epochs (batch size = 16, RMSprop optimizer), the model achieved:
  - Training Accuracy: ~90.4%
  - Validation Accuracy: ~85.5%
  - Validation Loss: ~0.43

This demonstrates a significant improvement over the Decision Tree, with a moderate train–validation gap.



[Figure 5.1: Training vs Validation Accuracy Graph]

B. Performance and Efficiency

The CNN model provides better accuracy by learning complex feature patterns, while the Decision Tree offers faster training and interpretability.

- Training Cost: CNN is computationally heavier (200 epochs)
- Inference Cost: Low — single forward pass
- Optimization: Joblib cac hing avoids repeated MFCC extraction

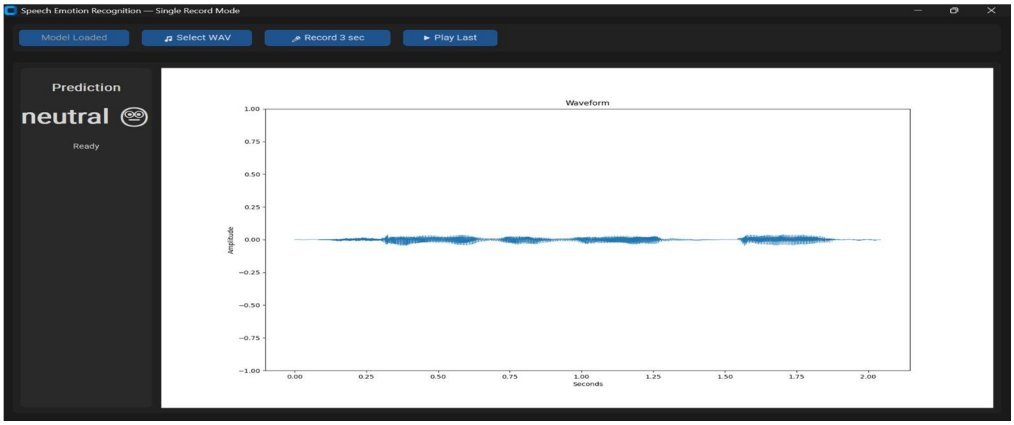
C. Real-Time Application (GUI Analysis)

The system includes a desktop-based inference application (LiveMic.py) that enables real-time emotion detection:

Workflow:

1) Load .h5 model (auto/manual)
2) Record 3-second audio or select WAV file
3) MFCC extraction → reshape to (1, 40, 1)
4) Model prediction using softmax
5) Display predicted emotion with waveform

The interface also supports audio playback and error handling for missing models.



[Figure 5.2: GUI Interface]

D. Limitations

1) Random train-test split may include same speakers → inflated accuracy
2) Acted datasets may not reflect real-world emotions
3) Mean-pooled MFCCs lose temporal information
4) Preprocessing mismatch (training uses gain, GUI does not)
5) Filename-based labeling is not robust
6) Model input shape must match (40,1)

E. Key Observation

The results clearly show that the CNN significantly outperforms the Decision Tree, validating the advantage of deep learning in capturing complex speech patterns. However, the system is best suited for demonstration and practical usage, rather than strict benchmarking without improved evaluation strategies.

VI. FUTURE SCOPE

The current system provides a functional SER pipeline; however, several improvements can enhance performance and practical usability:

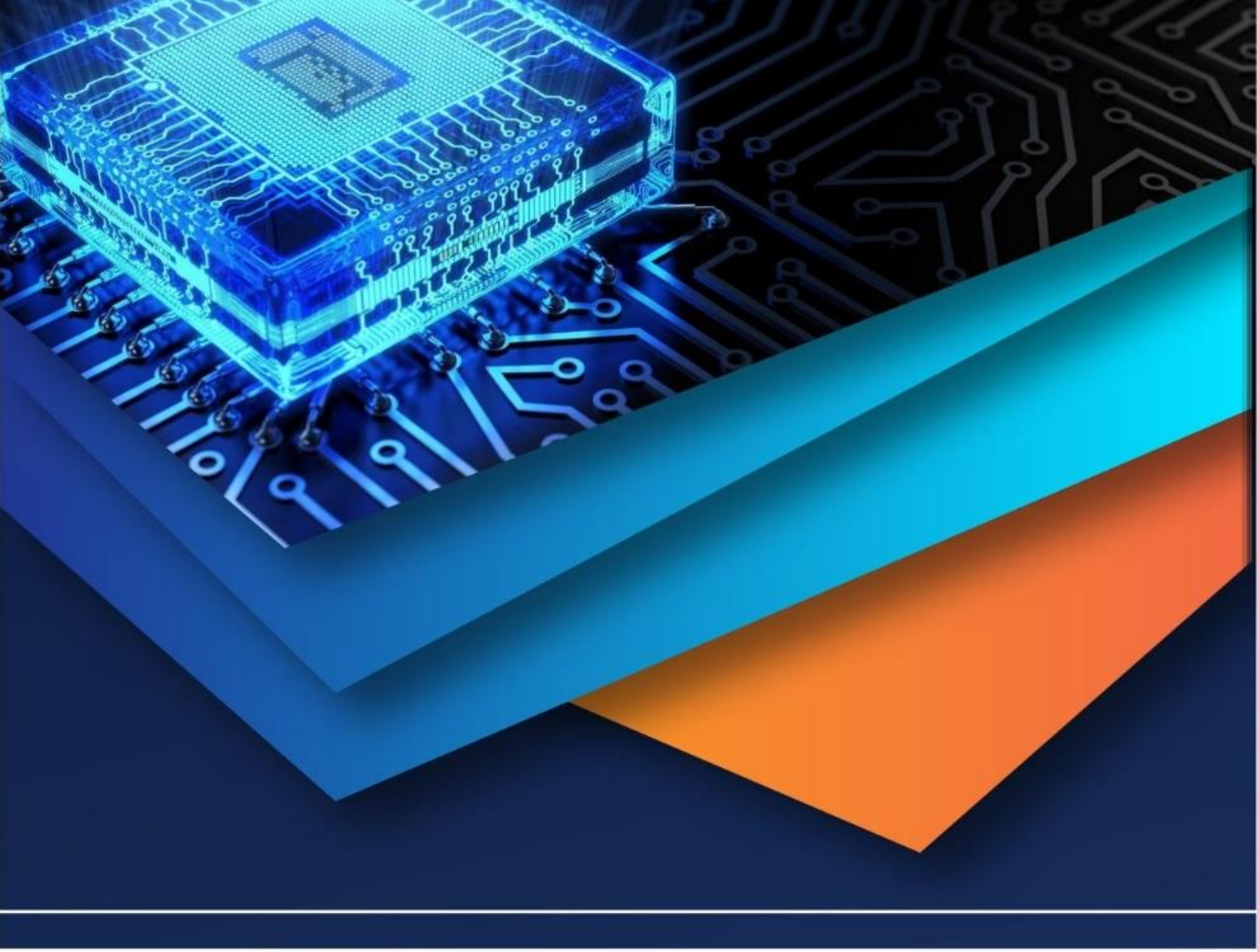
1) Speaker-Independent Validation: Future work should implement dataset splitting based on speaker identity to ensure no overlap between training and testing data. This will provide more realistic evaluation for unseen users.
2) Preprocessing Consistency: The preprocessing steps used during training and real-time inference should be aligned. Ensuring consistent gain, normalization, sampling rate, and MFCC extraction will improve prediction reliability.
3) Temporal Feature Modeling: Instead of using only mean-pooled MFCCs, future models can incorporate delta and delta-delta features or sequence-based inputs. This will help capture temporal variations in speech, improving emotion recognition accuracy.
4) Model Robustness Improvements: Techniques such as early stopping, learning rate scheduling, and regularization can be applied. Proper hyperparameter tuning will reduce overfitting and improve model stability.
5) Enhanced Inference and Reporting: The system can be extended to provide top-k predictions, confidence scores, and batch evaluation with CSV outputs. Additionally, exporting models to formats like ONNX or TensorFlow Lite can support deployment on lightweight or embedded systems.

VII. CONCLUSION

This project implements a Speech Emotion Recognition system using 40 MFCC features on a combined RAVDESS + TESS dataset. A Decision Tree model and a 1-D CNN were trained on the same data for comparison. The Decision Tree achieved around 68% accuracy, while the CNN reached approximately 85.5% validation accuracy, showing a clear improvement using deep learning on the same feature set. The system also includes a desktop application (LiveMic.py) for real-time prediction using microphone input or WAV files, along with waveform visualization and model loading support. Overall, the project delivers a working end-to-end pipeline covering feature extraction, model training, and real-time inference, with better performance observed from the CNN model.

REFERENCES

[1] S. R. Livingstone and F. A. Russo, “The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English,” PLOS ONE, vol. 13, no. 5, p. e0196391, 2018.
[2] K. Dupuis and M. K. Pichora-Fuller, “Toronto Emotional Speech Set (TESS),” University of Toronto, 2010. [Online]. Available: https://tspace.library.utoronto.ca/handle/1807/24487
[3] B. McFee et al., “librosa: Audio and music signal analysis in Python,” in Proc. 14th Python in Science Conference (SciPy), 2015, pp. 18–25.
[4] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[5] M. Abadi et al., “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015. [Online]. Available: https://www.tensorflow.org/
[6] F. Chollet et al., “Keras,” 2015. [Online]. Available: https://keras.io/
[7] S. Davis and P. Mermelstein, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-28, no. 4, pp. 357–366, Aug. 1980.
[8] A. Graves, A.-R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013, pp. 6645–6649.
[9] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proc. International Conference on Learning Representations (ICLR), 2015.
[10] T. N. Sainath et al., “Convolutional neural networks for LVCSR,” in Proc. IEEE ICASSP, 2013, pp. 8614–8618.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY