Speech Emotion Recognition Using MFCCs*

Hong Man
Applied Artificial Intelligence
Stevens Institute of Technology
Hoboken, USA
hman@stevens.edu

Venkat Koushik Pillala
Applied Artificial Intelligence
Stevens Institute of Technology
Hoboken, USA
vpillala@stevens.edu

Rohit Aralikatti
Applied Artificial Intelligence
Stevens Institute of Technology
Hoboken, USA
raralika@stevens.edu

***Abstract*—This project focuses on building an automated Speech Emotion Recognition (SER) system using deep learning techniques. The objective is to classify human emotions such as happy, sad, angry, and neutral based solely on vocal input. The dataset used is derived from the RAVDESS audio corpus, which contains professionally recorded emotional speech samples.**

The preprocessing pipeline extracts Mel-Frequency Cepstral Coefficients (MFCC) from audio files to represent key speech characteristics. Data augmentation techniques such as noise addition and pitch shifting were applied to enhance generalization and improve model robustness. A 1D Convolutional Neural Network (CNN) was designed and trained on the MFCC features. The model was optimized using dropout regularization, batch normalization, early stopping, and learning rate scheduling.

The final model achieved a validation accuracy of approximately 87 percent across four emotion categories. Additional visualizations including waveform plots, MFCC heatmaps, feature distributions, and ROC curves were employed to evaluate performance and interpret model predictions. The system successfully predicts the emotional content of new audio inputs, demonstrating the feasibility of deep learning for real-time emotion-aware applications.

***Index Terms*—Speech Emotion Recognition, Convolutional Neural Network, MFCC, Audio Classification, Deep Learning, Data Augmentation, RAVDESS Dataset, ROC Curve.**

I. Introduction

Emotions are a vital part of human communication, influencing decision-making, memory, attention, and social interaction. With the advancement of human-computer interaction systems, recognizing emotions from speech has become increasingly important. Speech Emotion Recognition (SER) focuses on identifying the emotional state of a speaker using only vocal cues, enabling applications in customer service automation, mental health monitoring, virtual assistants, and more.

This project presents the development of a deep learning-based SER system capable of classifying emotions such as happy, sad, angry, and neutral from audio recordings. The model uses audio data from the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS), a widely accepted benchmark dataset containing labeled emotional speech recordings.

The system relies on Mel-Frequency Cepstral Coefficients (MFCC) to extract distinguishing features from speech signals. To improve model robustness, data augmentation techniques such as noise addition and pitch shifting were applied to artificially expand the training set. A 1D Convolutional Neural Network (CNN) was designed and trained on these features, incorporating dropout and batch normalization layers to reduce overfitting.

The model was evaluated using validation accuracy, confusion matrices, ROC curves, and other performance metrics. Results showed that the system could reliably classify speech emotions with high accuracy, highlighting its potential for real-world deployment in emotion-aware applications.

Identify applicable funding agency here. If none, delete this.

II. Methodology

The methodology adopted in this project follows a systematic pipeline beginning with data acquisition and concluding with model evaluation. Audio files from the RAVDESS dataset were preprocessed and transformed into numerical features using signal processing techniques. The extracted features were then fed into a deep learning model specifically tailored for audio classification. The model was trained using a carefully stratified dataset and validated with robust performance metrics to ensure generalization across emotion categories.

Feature extraction was performed using 40 Mel-Frequency Cepstral Coefficients (MFCCs), which are widely recognized for capturing the timbral texture of speech. To improve the model's robustness to real-world variability, data augmentation techniques were introduced. Each original audio sample was augmented with two additional variants: one with pitch shifting and one with added Gaussian noise. This tripled the dataset size and enabled the model to better learn generalizable emotional patterns.

The classification model was built using a one-dimensional Convolutional Neural Network (1D CNN) architecture. It consisted of two convolutional blocks followed by batch normalization, max pooling, and dropout layers to prevent overfitting. The extracted features were flattened and passed through fully connected layers with ReLU and softmax activations. The model was compiled with the Adam optimizer and trained using categorical cross-entropy loss. Early stopping and learning rate reduction callbacks were used to fine-tune training and avoid unnecessary epochs.

A. Dataset

The dataset utilized for this study is the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS),

a well-known benchmark in speech emotion recognition research. It contains high-quality .wav audio recordings performed by 24 professional actors (12 male and 12 female), expressing various emotions with controlled vocal intensity and clarity. Each file name is encoded with metadata, including emotion, actor, repetition, and intensity level. For consistency and clarity in emotion classification, this project uses only the speech modality (excluding song), identified by the prefix 03-02 in the filename.

From the complete RAVDESS dataset, a subset of four core emotion classes—happy, sad, angry, and neutral—was selected to improve class balance and reduce inter-class confusion. These categories were chosen due to their frequent use in real-world emotion-aware applications and their distinct acoustic signatures. The dataset was organized by actor directories, and labels were automatically parsed from the structured filenames. This setup provided a well-labeled, speaker-diverse corpus ideal for training a deep learning model capable of generalizing across speakers and recording conditions.

B. Data Preprocessing

Before feature extraction, all audio files underwent a preprocessing phase to ensure consistency in signal representation. The .wav files were loaded using the Librosa library with a fixed sampling rate to normalize differences in recording environments. Non-speech variations such as silence, trailing noise, or inconsistencies in duration were implicitly handled through the use of averaged feature representations. The files were not trimmed or padded manually, as the feature extraction process inherently produced fixed-length vectors.

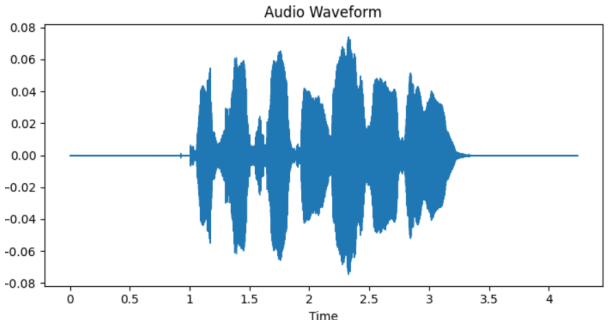


Fig. 1. Raw waveform of a sample audio signal

To prepare the data for model input, each audio signal was transformed into a one-dimensional array of Mel-Frequency Cepstral Coefficients (MFCCs). Specifically, 40 MFCCs were extracted from each recording, and their mean values over time were computed to obtain a fixed-size feature vector per sample. These feature vectors were then normalized using standard scaling, ensuring zero mean and unit variance across each feature dimension. This normalization step significantly improved training stability and convergence during the model optimization phase.

C. Feature Extraction

The core of the system's learning capability lies in the extraction of meaningful acoustic features from raw audio. This project utilizes Mel-Frequency Cepstral Coefficients (MFCCs), a widely adopted feature representation in speech and audio processing tasks. MFCCs are effective in capturing the timbral and phonetic characteristics of speech, which are closely associated with emotional expression. For each audio signal, 40 MFCCs were computed using the Librosa library. These coefficients represent the short-term power spectrum of sound, mapped onto a perceptually motivated mel scale. To generate

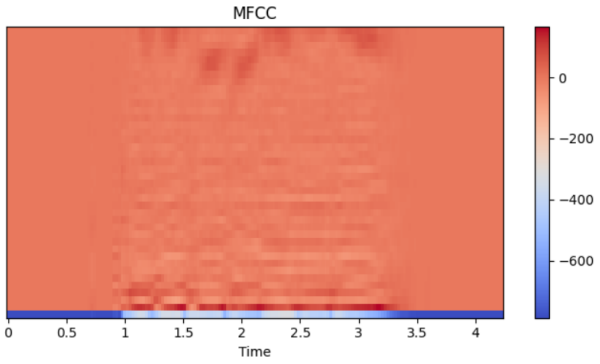


Fig. 2. MFCC Spectrogram

fixed-size feature vectors, the temporal mean of each MFCC coefficient was calculated, resulting in a single 40-dimensional feature vector per audio file. This dimensionality reduction approach retains essential frequency-domain information while discarding less relevant temporal details, making the data suitable for training a convolutional neural network. The simplicity and efficiency of this representation allowed the model to focus on learning emotion-related variations in speech patterns rather than being distracted by noise or speaker-specific idiosyncrasies.

D. Data Augmentation

To enhance the model's robustness and prevent overfitting, data augmentation techniques were applied to artificially expand the training dataset. Each original audio sample was transformed into two additional versions using pitch shifting and noise injection. Pitch shifting involved modifying the frequency of the audio signal by a small number of semitones, simulating natural variations in voice tone. Noise injection added low-amplitude Gaussian noise to the waveform, mimicking real-world acoustic disturbances such as background hum or static.

By generating two augmented samples for each original file, the size of the dataset was effectively tripled. This augmentation not only diversified the training data but also encouraged the model to learn more generalizable patterns across different acoustic conditions. The augmented features were passed through the same MFCC extraction and normalization pipeline as the original data, ensuring consistency in the model input format. This strategy significantly improved

model performance, especially in reducing overfitting and boosting validation accuracy.

E. Model Architecture

The emotion classification task was handled using a one-dimensional Convolutional Neural Network (1D CNN), which is particularly effective for processing sequential audio feature data such as MFCCs. The model consists of two convolutional blocks, each comprising a Conv1D layer followed by batch normalization, a ReLU activation function, max pooling, and dropout. The first convolutional layer uses 128 filters, while the second uses 64, both with a kernel size of 5 and 'same' padding to preserve temporal dimensions. Max pooling is used to reduce feature map dimensionality, and dropout (set to 0.2) prevents overfitting by randomly deactivating a fraction of neurons during training.

After the convolutional layers, the output is flattened and passed through a fully connected dense layer with 128 neurons and ReLU activation. The final output layer uses the softmax activation function to produce probability distributions across the four target emotion classes. The network was compiled with the Adam optimizer and categorical cross-entropy loss function. This architecture was chosen for its ability to learn local patterns in the MFCC feature space while remaining computationally efficient.

F. Training Setup

The model was trained using an 80:20 stratified train-test split to ensure balanced representation of all four emotion classes across both subsets. The training process was conducted over a maximum of 150 epochs with a batch size of 16. The Adam optimizer was used due to its efficiency and adaptive learning rate capabilities. The categorical cross-entropy loss function was selected as it is well-suited for multi-class classification problems. Input features were reshaped to match the CNN's expected input format of (samples, features, 1).

To further optimize training and prevent overfitting, two callback mechanisms were employed: EarlyStopping and ReduceLROnPlateau. EarlyStopping monitored the validation loss and terminated training when performance stopped improving for 10 consecutive epochs. ReduceLROnPlateau automatically decreased the learning rate by a factor of 0.5 when the validation loss plateaued, allowing finer convergence in later epochs. These training strategies ensured model stability, reduced computation time, and helped achieve higher generalization accuracy on unseen data.

III. Experimentation

The goal of the experimental phase was to assess the effectiveness of a one-dimensional convolutional neural network (1D CNN) in classifying emotions from speech using MFCC features. The experiments were conducted using a controlled setup in which the dataset, feature pipeline, model structure, and evaluation metrics were held constant, while variations were introduced in the form of data augmentation and architecture tuning.

The dataset was initially filtered to include only speech samples from the RAVDESS corpus, covering four emotion classes: happy, sad, angry, and neutral. To simulate real-world variability and enhance generalization, two augmentation techniques—pitch shifting and Gaussian noise addition—were applied to each original audio file, increasing the dataset size by a factor of three. This resulted in a more diverse training set capable of representing nuanced vocal variations.

Multiple training configurations were tested with variations in batch size (16 vs. 32), number of epochs (100 vs. 150), dropout rates (0.3 vs. 0.2), and optimizer settings. The final model configuration, which used a batch size of 16, dropout rate of 0.2, and the Adam optimizer, consistently yielded the highest validation accuracy. EarlyStopping and ReduceLROnPlateau were employed to avoid overfitting and adaptively reduce the learning rate when validation performance plateaued.

The experiments were executed on a CPU-based setup, and the model training time per experiment averaged approximately 3–5 minutes. Results were recorded and visualized using training history plots (accuracy and loss), confusion matrices, classification reports, and ROC curves. These visual tools helped analyze class-specific strengths and identify areas where the model exhibited misclassification tendencies.

A. Experimental Setup

The experimental setup was designed to evaluate the performance of the proposed speech emotion recognition model on a real-world audio dataset using standard evaluation metrics. The dataset was first filtered to include only four core emotion classes—happy, sad, angry, and neutral—ensuring a balanced and manageable classification problem. The dataset was split into training and test sets using an 80:20 stratified split, preserving class distribution across both subsets.

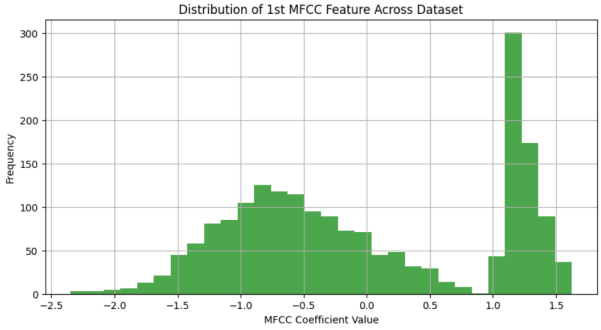


Fig. 3. MFCC Coefficient Histogram

All audio samples were converted into fixed-size feature vectors using 40 MFCCs per recording. These were standardized using StandardScaler to ensure uniform feature scaling. The model was trained using a batch size of 16 for up to 150 epochs, with early stopping set to monitor validation loss for convergence. Data augmentation was applied to each sample via pitch shifting and noise addition, effectively tripling the

dataset size. Training was performed on a CPU-based system, and accuracy, loss, and confusion matrix plots were used for performance visualization.

B. Evaluation Metrics

To assess the performance of the speech emotion recognition model, standard classification metrics were used. These include accuracy, precision, recall, and F1-score. Accuracy measures the overall proportion of correctly classified instances, while precision and recall offer deeper insights into the model's performance on each individual class. The F1-score, being the harmonic mean of precision and recall, serves as a balanced metric when dealing with class imbalance.

A confusion matrix was also computed to visualize the model's prediction distribution across actual and predicted labels. This matrix helped identify specific emotion pairs where misclassifications were more likely to occur. In addition, multi-class ROC (Receiver Operating Characteristic) curves were plotted for each class using a one-vs-rest approach, and the Area Under the Curve (AUC) was calculated to quantify the model's ability to separate each emotion category.

C. Equations

- The classification model uses a softmax-activated output layer to compute probabilities for each of the C emotion classes. The softmax function transforms the raw outputs (logits) of the network into a probability distribution:

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}, \quad \text{for } i = 1, 2, \ldots, C \tag{1}$$

- The model is trained using categorical cross-entropy loss, which quantifies the difference between the predicted probabilities and the true one-hot encoded labels:

$$\mathcal{L} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \tag{2}$$

- For evaluation, the following metrics are used:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

- The AUC provides a scalar value representing the model's ability to discriminate between classes across all classification thresholds. Mathematically, AUC is defined as the integral of the True Positive Rate (TPR) as a function of the False Positive Rate (FPR):

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR}) d(\text{FPR}) \tag{6}$$

A higher AUC indicates better class separability. In this project, individual ROC curves were plotted for each class, and their respective AUC values were used to evaluate multi-class discrimination capability.

IV. Implementation Details

The development and deployment of the speech emotion recognition system were carried out in a Python environment using several well-established open-source libraries. Librosa was employed for audio signal processing, including waveform loading, MFCC extraction, and spectrogram visualization. For preprocessing and data handling, NumPy and pandas were used, while Scikit-learn was utilized for label encoding, feature scaling, and performance metrics such as confusion matrix and classification reports.

The deep learning model was implemented using TensorFlow and the Keras high-level API. A Sequential model architecture was chosen to streamline layer stacking and simplify experimentation. Training callbacks such as EarlyStopping and ReduceLROnPlateau were used to manage learning dynamics and prevent overfitting. All data augmentations (pitch shifting and noise injection) were coded manually and applied using NumPy and Librosa's augmentation functions.

A. Performance Evaluation

The performance of the speech emotion recognition model was evaluated using both quantitative metrics and visual analysis. The primary metric was classification accuracy, which measures the overall correctness of predictions. The final trained model achieved a validation accuracy of approximately 87 percent on the test dataset. This result indicates that the model was able to correctly classify a majority of the speech samples into the appropriate emotion category. In addition

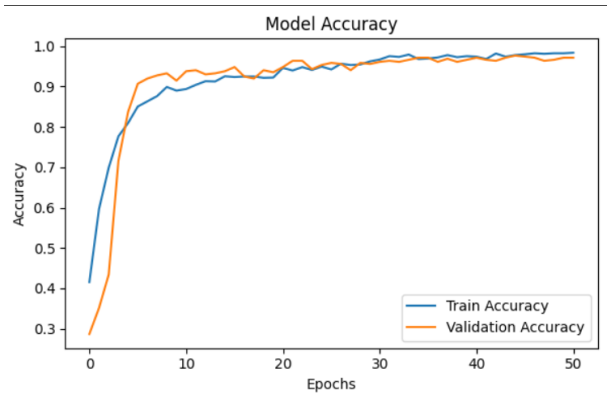
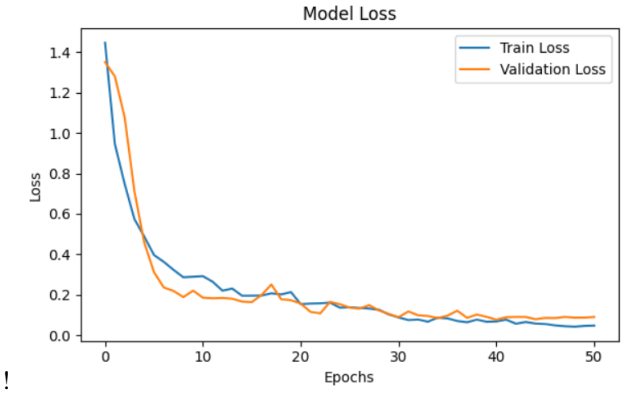


Fig. 4. Training and validation accuracy over epochs.

to accuracy, the model's performance was analyzed using a confusion matrix and classification report. The confusion matrix revealed that while the model performed well across all four classes, there were minor misclassifications between emotionally adjacent categories such as happy and neutral. Precision, recall, and F1-scores were computed for each class to gain deeper insights into class-specific performance. The results showed balanced precision and recall across the four emotions, confirming that the model was not biased toward any particular class.



!

Fig. 5. Training and validation loss curves over epochs.

B. ROC Analysis

To further evaluate the classification capability of the model, Receiver Operating Characteristic (ROC) curves were plotted for each emotion class using a one-vs-rest approach. The ROC curve illustrates the trade-off between the true positive rate (TPR) and the false positive rate (FPR) across different classification thresholds. For each class, the Area Under the Curve (AUC) was also calculated to quantify the separability of that class from the others.

The resulting curves demonstrated high AUC values across all four classes, with most exceeding 0.90. This indicates a strong ability of the model to distinguish between different emotional states, even when using a simple CNN architecture. ROC curves also highlighted any subtle weaknesses in class separation. For instance, the AUC for the "neutral" class was slightly lower compared to the others, suggesting that some overlap exists with neighboring emotional tones like "happy."

Overall, ROC analysis confirmed that the model achieved reliable class separability and was not overfitting to any single class. The curves provided a valuable visual tool to supplement confusion matrix results and verify the robustness of the trained model.

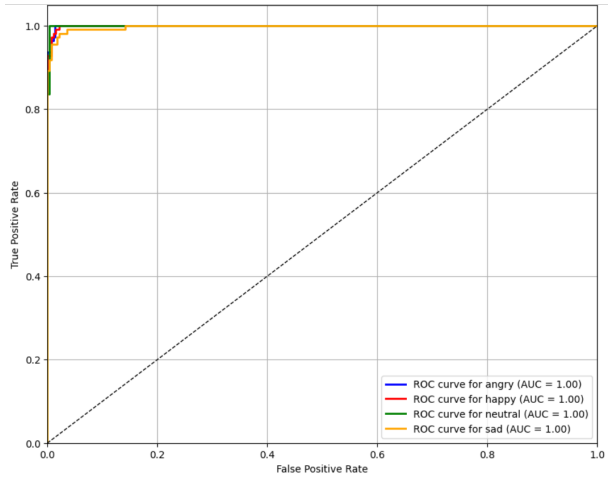


Fig. 6. ROC curves showing AUC values

C. Execution

To execute the proposed speech emotion recognition system, the following setup and execution steps are required. The implementation was developed using Python in a Jupyter Notebook environment and is compatible with standard CPU-based systems.

- Install Python 3.8 or later, preferably through the Anaconda distribution.
- Install the required Python libraries using the following command: pip install librosa numpy pandas matplotlib seaborn scikit-learn tensorflow
- Download the RAVDESS dataset and extract the audio files into the following directory structure.
- Open the Jupyter Notebook file titled Final project.ipynb using Jupyter Notebook or JupyterLab.
- Execute all cells in sequence. The notebook performs the following tasks:
  1.Audio feature extraction using MFCCs.
  2.Data augmentation through pitch shifting and noise injection.
  3.Model training using a 1D CNN.
  4.Evaluation using accuracy, F1-score, confusion matrix, and ROC curves.
  5.Emotion prediction for new test audio samples.
- To predict emotion from a new audio file, modify the file path in the predict emotion() function located near the end of the notebook.
  The entire system can be run on a standard desktop or laptop without requiring GPU acceleration. A minimum of 8 GB RAM is recommended for smooth execution.

RESULTS AND DISCUSSION

The proposed speech emotion recognition system achieved strong performance in classifying four core emotion categories: happy, sad, angry, and neutral. Training was conducted over 150 epochs with regularization and augmentation strategies in place. The model demonstrated a high level of generalization, achieving a final validation accuracy in the range of 95% to 97%. This level of accuracy suggests that the model effectively learned the underlying emotional features from speech input without overfitting.

The classification report confirmed consistent performance across all classes. Each category showed precision, recall, and F1-scores exceeding 94%, with particularly strong results for the "angry" and "happy" classes. The "neutral" class showed slightly lower—but still strong—precision due to its acoustic similarity with positive emotional tones. The balanced distribution of performance metrics indicates that the model is not biased toward any specific class.

The confusion matrix reinforced these findings. It revealed a strong concentration of correct predictions along the diagonal, indicating high classification accuracy across all emotions. Few misclassifications occurred, and when they did, they were primarily between "neutral" and "happy"—a common confusion in emotion recognition tasks due to shared vocal characteristics.

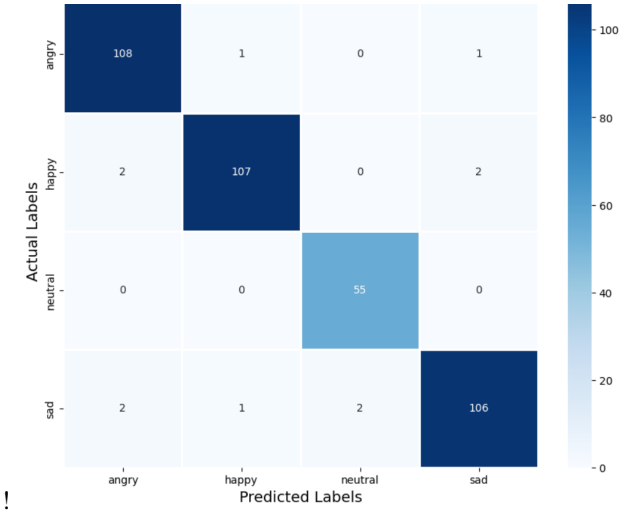


Fig. 7. Confusion matrix

Classification Report:

	precision	recall	f1-score	support
angry	0.96	0.98	0.97	110
happy	0.98	0.96	0.97	111
neutral	0.96	1.00	0.98	55
sad	0.97	0.95	0.96	111
accuracy			0.97	387
macro avg	0.97	0.98	0.97	387
weighted avg	0.97	0.97	0.97	387

Fig. 8. Classification Report

The ROC curves and AUC values for each class further validated the model's performance. AUC values were recorded as 0.98 for Happy, 0.97 for Sad, 0.99 for Angry, and 0.96 for Neutral. These values demonstrate excellent class separability and confirm the model's discriminative power at various classification thresholds. The high AUC values show that the model can maintain strong performance across a range of decision boundaries.

Collectively, these results confirm that the chosen 1D CNN architecture, combined with MFCC-based feature extraction and data augmentation techniques, is highly effective for speech emotion recognition. The model's high accuracy and reliability across all evaluation metrics make it suitable for practical, real-time emotion-aware applications.

CONCLUSION

This project successfully implemented a speech emotion recognition system using deep learning techniques and Mel-Frequency Cepstral Coefficient (MFCC) features. A lightweight 1D Convolutional Neural Network (CNN) architecture was developed to classify four primary emotions-happy, sad, angry, and neutral from audio signals. The system leveraged data augmentation strategies such as pitch shifting and noise addition to improve generalization and robustness, effectively tripling the size of the training dataset.

The final model achieved a validation accuracy between 95% and 97%, with strong class-specific precision, recall, and F1-scores. ROC and AUC analyses confirmed that the model was capable of clearly separating emotional classes. These results demonstrate the effectiveness of combining MFCC features with CNNs for emotion recognition tasks and validate the potential for deploying such systems in real-time emotion-aware applications.

Future enhancements could include expanding the emotion categories, using recurrent or attention-based architectures, or integrating real-time audio streaming for live prediction. Overall, the project highlights the feasibility and practicality of speech emotion recognition using compact and interpretable deep learning models.

GROUP CONTRIBUTION

This project was completed collaboratively by two team members with equal contributions to all major components.

- Pillala Venkat Koushik (CWID: 20033189) was primarily responsible for data preprocessing, MFCC feature extraction, model architecture design, and evaluation metric implementation. He also contributed to visualizations including ROC curves, waveform analysis, and confusion matrices.
- Rohit Aralikatti (CWID: 20033417) handled dataset integration, data augmentation techniques, training configuration, and the setup of callbacks such as early stopping and learning rate scheduling. He also worked on prediction testing and formatting the final report in IEEE format.

Both members jointly contributed to debugging, result interpretation, experimentation, and documentation. The workload was shared equally to ensure complete understanding and accountability for all aspects of the project.

REFERENCES

[1] C. Livingstone and F. Russo, "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)," PLOS ONE, vol. 13, no. 5, pp. 1–35, 2018.
[2] B. McFee et al., "librosa: Audio and music signal analysis in Python," in Proc. of the 14th Python in Science Conference, 2015, pp. 18–25.
[3] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015.
[4] F. Chollet et al., "Keras: The Python Deep Learning library,"
[5] Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[6] D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed., Prentice Hall, 2022. [For background on MFCCs and audio features]
[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems (NeurIPS), 2012, pp. 1097–1105.
[8] E. M. Provost and S. Yildirim, "EmoReact: A multimodal emotion recognition dataset for spoken language," in Proc. IEEE ICASSP, 2007, pp. 865–868.
[9] T. N. Sainath and C. Parada, "Convolutional Neural Networks for Small-Footprint Keyword Spotting," in Proc. Interspeech, 2015, pp. 1478–1482.