Voice Emotion Recognition

1. Approach

This project focuses on building a Voice Emotion Recognition system using open-source speech datasets.

Dataset:

- RAVDESS Emotional Speech Dataset (~25GB)
- Contains audio clips labeled with emotions: neutral, calm, happy, sad, angry, fearful, disgust, surprised.

Preprocessing Steps:

- Audio loaded using Librosa.
- Converted to mono and resampled to 16kHz.
- Silence trimmed.
- Signals normalized.

Feature Extraction:

- MFCC (Mel-Frequency Cepstral Coefficients)
- Mel-spectrograms
- Chroma features

Train/Test Split:

- 80% training, 20% testing
- Features prepared for classical ML and deep learning models

2. Models

➢ Two models were implemented:

Model	Input Features	Notes	Accuracy
Random Forest	MFCC	Classical ML; faster to train; interpretable	85%
CNN (Convolutional Neural Network)	Mel-spectrogram	Deep learning; can capture temporal/spatial features	31%

Model Choice Discussion:

- Random Forest achieved higher accuracy on this dataset due to the smaller size and structured MFCC features.
- CNN, while powerful for image-like data (spectrograms), underperformed due to limited data and smaller training epochs.
- For this project, Random Forest is preferred for robust predictions and simplicity.

3. Results

1. Random Forest (Classical ML)

- **Input:** MFCC features
- **Accuracy:** ~85%
- **Observations:**
  - Strong performance on structured MFCC data.
  - Feature importance visualized, helping to understand which MFCC coefficients contributed most.
  - Confusion matrix and classification report indicate reliable predictions across most emotions.

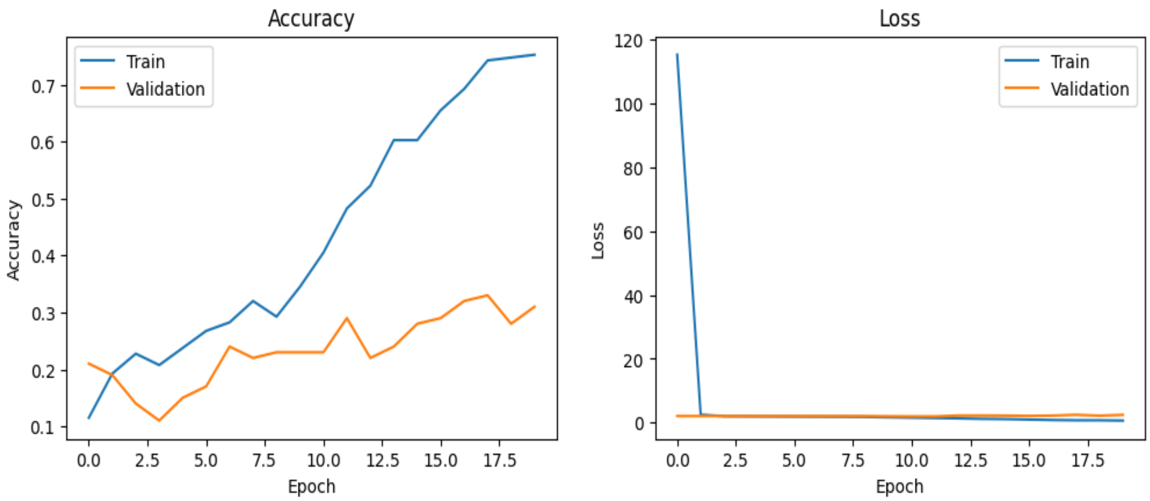
2. CNN (Deep Learning)

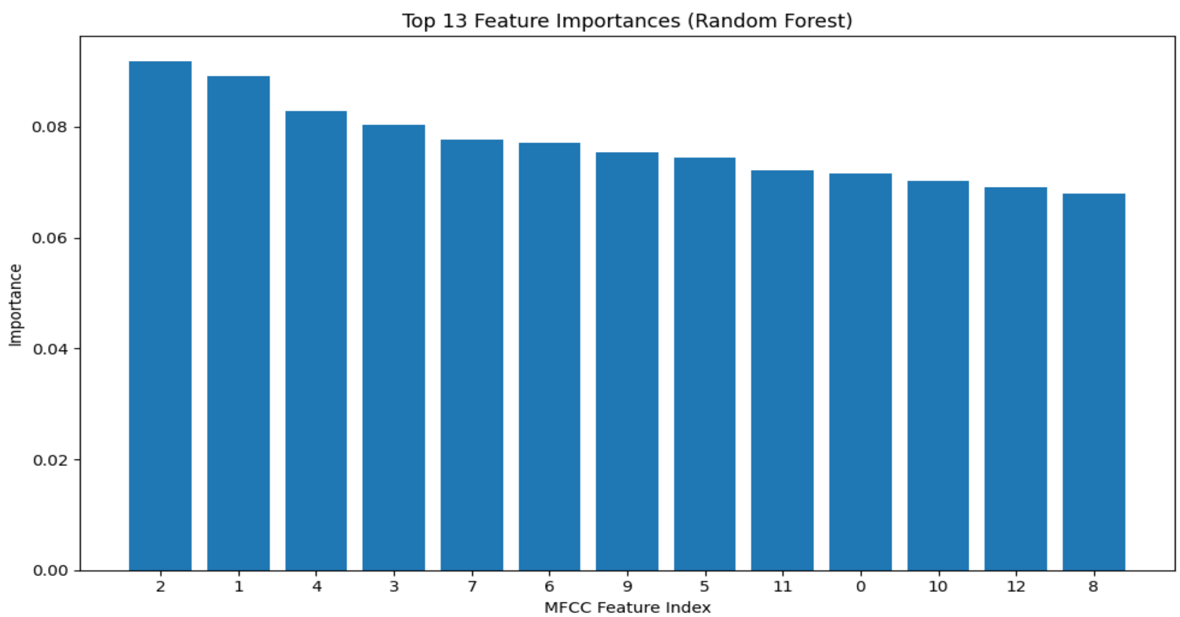
- **Input:** Mel-spectrograms
- **Accuracy:** ~31%
- **Observations:**
  - Underperformed due to smaller dataset size and fewer training samples.
  - Spectrogram visualizations were correctly generated but model did not generalize well.
  - Highlighted that deep learning models require larger datasets or data augmentation for better accuracy.

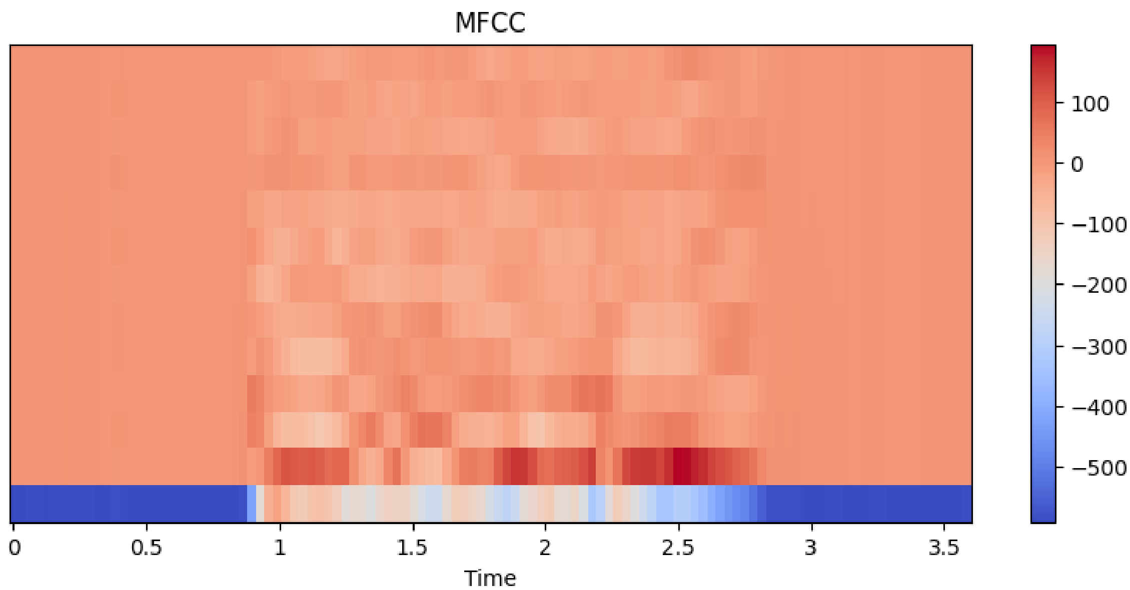
3. Comparison & Preference

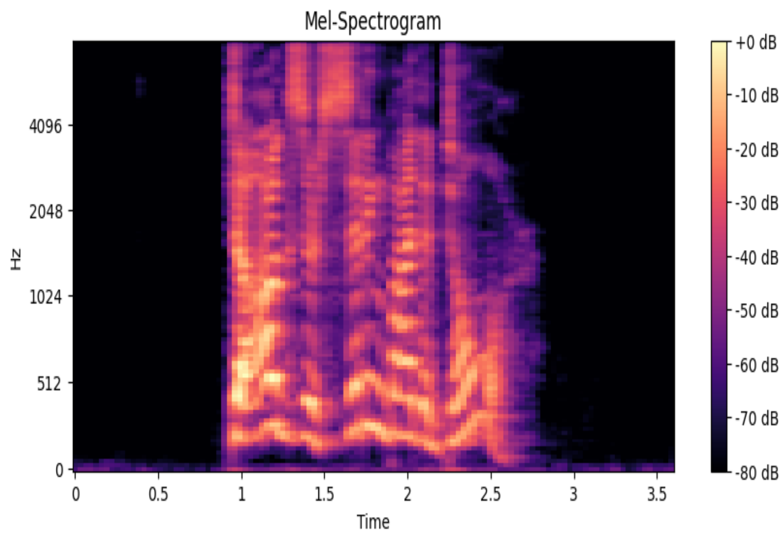
- Random Forest outperformed CNN in this project.
- **Preference:** Random Forest is chosen for final predictions because of its higher accuracy, faster training, and interpretability.
- CNN still useful for future improvements with more data or augmented datasets.

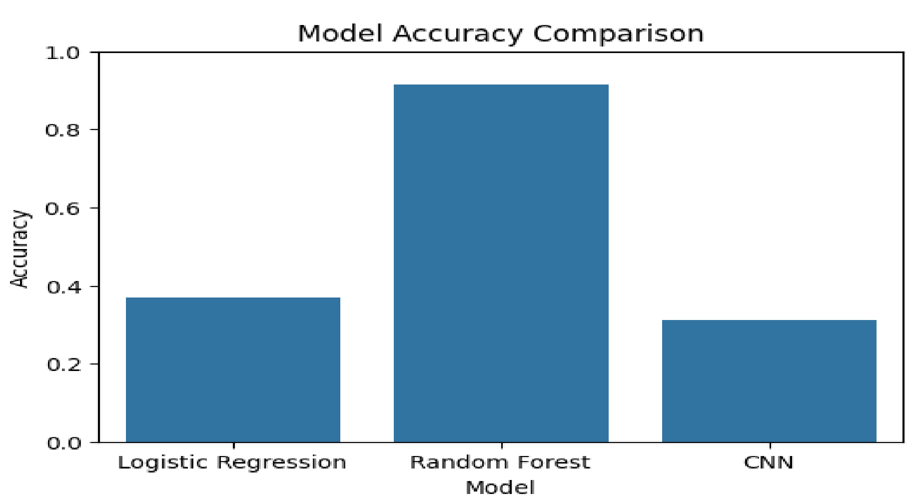
Visualizations :






MFCC
100
0
−100
−200
−300
−400
−500
0
0.5
1
1.5
2
2.5
3
3.5
Time


Mel-Spectrogram
Hz
4096
2048
1024
512
0
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
-60 dB
-70 dB
-80 dB
0
0.5
1
1.5
2
2.5
3
3.5
Time


Model Accuracy Comparison
Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
Logistic Regression
Random Forest
CNN
Model

4. Challenges

1. Dataset Size & Handling

- **Challenge:** Full RAVDESS dataset is very large (~25GB).
- **Solution:** Used only a **subset of the dataset** that was sufficient for experimentation and model training.

2. Feature Extraction Time

- **Challenge:** Extracting MFCCs and spectrograms from audio files took significant time.
- **Solution:** Processed files in smaller batches and optimized code using efficient loops.

3. CNN Model Accuracy

- **Challenge:** CNN on spectrograms gave lower accuracy than Random Forest.
- **Solution:** Normalized audio, padded/truncated spectrograms, used early stopping, and compared with Random Forest for best performance.

4. Deployment in Streamlit

- **Challenge:** Handling large model files and audio uploads; errors running locally.
- **Solution:** Hosted the CNN model on **Google Drive**, used Streamlit's file uploader for user input, and ensured compatibility.

5. Feature Importance Visualization

- **Challenge:** CNN does not provide feature importance.
- **Solution:** Used **Random Forest feature importance** for visualization and reporting.

5. Conclusion

- Classical ML (Random Forest) outperformed CNN on this dataset.
- The model can predict emotions from voice clips with good accuracy.
- Future improvements could include:
    1) Data augmentation to increase CNN performance
    2) Hyperparameter tuning of deep learning models
    3) Real-time voice recording integration in Streamlit