Emotion Recognition from Speech

Project Overview

The goal of this project is to build a system that can recognize human emotions (such as happy, sad, angry, etc.) from speech audio using deep learning and speech signal processing techniques.

Key Components

1. Feature Extraction: MFCC (Mel-Frequency Cepstral Coefficients)

- **What:** MFCCs represent the short-term power spectrum of sound and are widely used in audio processing tasks.
- **Why:** They capture the characteristics of speech that are crucial for distinguishing between different emotions.
- **How:** Using the `librosa` library in Python to extract MFCC features from audio files.

```
import librosa
import numpy as np

file_path = 'your_audio.wav'
y, sr = librosa.load(file_path, duration=3, offset=0.5)
mfccs = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=40)
mfccs_scaled = np.mean(mfccs.T, axis=0)
```

Deep Learning Models Used

2. Convolutional Neural Networks (CNN)

- **What:** A type of deep learning model originally designed for images but also effective for processing audio represented as 2D features (like MFCCs).
- **Why:** CNNs can automatically learn spatial patterns in the MFCCs.
- **How:** Using layers of convolutions and pooling to extract and condense features.

Algorithm:

1. Input: 2D MFCC feature map.
2. Apply convolutional layers to extract patterns.
3. Apply pooling to reduce dimensionality.
4. Flatten and pass through dense layers for classification.

```
model_cnn = Sequential([...])
```

---

3. Recurrent Neural Networks (RNN)

- **What:** A type of neural network specifically designed for sequential data.
- **Why:** Speech is inherently sequential; RNNs can capture the temporal dynamics of spoken words.
- **How:** By maintaining a hidden state that evolves over time as it reads the sequence.

Algorithm:

1. Input: Sequence of MFCC features.
2. Feed into RNN cells (tanh activations).
3. Capture temporal patterns.
4. Dense layers for emotion classification.

```
model_rnn = Sequential([...])
```

---

4. Long Short-Term Memory Networks (LSTM)

- **What:** A special type of RNN designed to overcome the vanishing gradient problem in standard RNNs.
- **Why:** Better at capturing long-term dependencies and context in speech data.
- **How:** Uses gates to control the flow of information.

Algorithm:

1. Input: Sequence of MFCC features.
2. Process through LSTM cells with input, forget, and output gates.
3. Dense layers for classification.

```
model_lstm = Sequential([...])
```

Datasets Used

- **RAVDESS:** Ryerson Audio-Visual Database of Emotional Speech and Song.
- **TESS:** Toronto Emotional Speech Set.
- **EMO-DB:** Berlin Emotional Speech Database.

Summary Table

Model	Strength	Weakness
CNN	Captures local patterns in features	May miss temporal context
RNN	Models sequences and time dependencies	Can suffer from vanishing gradients
LSTM	Remembers long-term dependencies	Computationally more expensive

---

Conclusion

By combining MFCC feature extraction with powerful deep learning models like CNN, RNN, and LSTM, we can effectively classify emotions from speech signals. Each model offers unique advantages and can be selected based on the specific requirements of accuracy, speed, and computational resources.

> Note: All implementations are done using Python with libraries such as Librosa, TensorFlow/Keras.