`**INTRODUCTION**

The Speech Emotion Recognition (SER) project aims to develop a system capable of accurately detecting and classifying emotions from audio recordings of human speech. Emotion recognition has significant applications in various fields such as human-computer interaction, customer service, and healthcare. By accurately identifying the emotional state of the speaker, the system can provide valuable insights and enhance user experiences.

**PROBLEM DEFINITION & FEASIBILITY ANALYSIS**

The primary challenge addressed by this project is the accurate recognition and classification of emotions from audio samples. Emotion recognition from speech signals is a complex task due to variations in speech patterns, accents, and emotional expressions. However, recent advancements in machine learning and signal processing techniques have made it feasible to develop robust SER systems.

**SOFTWARE REQUIREMENTS SPECIFICATION**

**SYSTEM REQUIREMENTS:**

**HARDWARE SPECIFICATION:**

1. **Processor (CPU):**
   * Dual-core processor or higher.
   * Recommended: Intel Core i5 or Ryzen 5.
2. **Memory (RAM):**
   * Minimum: 4 GB RAM.
   * Recommended: 8 GB RAM or higher for better performance.
3. **Storage Space:**
   * At least 3 GB of free disk space for storing datasets, audio files, and project files.
   * Additional space may be required depending on the size of datasets and generated files.

**SOFTWARE SPECIFICATION:**

1. **Operating System:**

Windows 10, macOS, or Linux-based operating system.

1. **Python**

Ensure you have Python installed on your system. You can download and install Python from the official Python website <https://www.python.org/downloads/>

1. **Python libraries:**

pandas: For data manipulation and handling Excel files.

scikit-learn: For machine learning algorithms and evaluation metrics.

NumPy: For numerical operations.

librosa: For audio feature extraction.

pydub: For audio processing and manipulation.

SciPy: For signal processing and filtering.

Keras with TensorFlow backend: For building and training deep learning models.

seaborn: For statistical data visualization based on matplotlib.

matplotlib: For creating static, animated, and interactive visualizations in Python.

**Install these libraries using -** pip install pandas scikit-learn numpy librosa pydub scipy keras tensorflow seaborn matplotlib

1. **Text editor:**

Vs code

**USER REQUIREMENTS**

**FUNCTIONAL REQUIREMENTS:**

**Dataset:**

(RAVDESS) Ryerson Audio-Visual Database of Emotional Speech and Song is used as dataset for this

**Performance Requirement:**

This software will require good internet connection to connect with servers and a good processing system to give best performance

**Modules:**

1. **Noise Reduction:** This module preprocesses audio data by applying a low-pass filter to reduce background noise, improving the quality of audio recordings.
2. **Feature Extraction:** Extracts relevant features from audio signals, such as Mel-Frequency Cepstral Coefficients (MFCCs), which are used as input for emotion classification.
3. **Feature Scaling:** Normalizes or scales extracted features to ensure consistency and improve the performance of machine learning models.
4. **Data Splitting:** Splits the dataset into training and testing sets to evaluate model performance.
5. **Classifier Selection:** Chooses appropriate classifiers such as Support Vector Machines (SVM), Convolutional Neural Networks (CNNs), or Long Short-Term Memory (LSTM) networks for emotion classification.
6. **Model Training:** Trains the selected classifier using the training data to learn patterns and relationships between input features and emotion labels.
7. **Model Evaluation:** Evaluates the trained model's performance using the testing data, calculating metrics such as accuracy, precision, recall, and F1-score.

**4. System Design**

The system architecture consists of several interconnected modules:

* **Noise Reduction Module:** Preprocesses audio data to remove background noise.
* **Feature Extraction Module:** Extracts MFCC features from preprocessed audio samples.
* **Feature Scaling Module:** Normalizes the extracted features to ensure consistency.
* **Data Splitting Module:** Splits the dataset into training and testing sets.
* **Classifier Selection Module:** Selects appropriate machine learning classifiers for emotion classification.
* **Model Training Module:** Trains the selected classifier using the training data.
* **Model Evaluation Module:** Evaluates the trained model's performance using the testing data.

**5. Implementation**

**Module Implementation:**

* **Noise Reduction:** Implemented using a low-pass filter to remove background noise from audio recordings.
* **Feature Extraction:** Utilizes the Librosa library to extract MFCC features from audio samples.
* **Feature Scaling:** Employs the StandardScaler from scikit-learn to scale extracted features.
* **Data Splitting:** Utilizes train\_test\_split from scikit-learn to split the dataset into training and testing sets.
* **Classifier Selection:** Supports various classifiers such as SVM, CNNs, and LSTM networks.
* **Model Training:** Trains the selected classifier using the training data.
* **Model Evaluation:** Evaluates the trained model's performance using testing data and calculates classification metrics.

**6. Conclusion**

The SER project demonstrates the feasibility of accurately recognizing and classifying emotions from speech signals using machine learning techniques. By preprocessing audio data, extracting relevant features, and training robust classifiers, the system achieves high accuracy in emotion classification. The project lays the foundation for future research and applications in emotion-aware computing, human-computer interaction, and affective computing.

This documentation provides an overview of the project's objectives, methodology, implementation details, and outcomes, serving as a comprehensive guide for understanding and replicating the SER system.