**Project Documentation: Audio Analyzer with Emotion, Gender Classification, and Speech Diarization**

This project builds an audio processing pipeline that takes an audio file, segments it into speaker-specific parts using a diarization model, and then analyzes each segment for both emotion and gender. The system is composed of several modules:

1. **Audio Analyzer** (Main Pipeline)
2. **Audio Preprocessing**
3. **Emotion Classification**
4. **Gender Classification**
5. **Speech Diarization**

Additionally, the solution leverages pre-trained models for speaker embedding extraction and diarization.

**1. Technologies and Libraries Used**

* **Python** – The primary programming language.
* **PyTorch & Torch.nn** – For constructing and running deep neural network models.
* **Librosa** – For audio loading, resampling, and feature extraction (mel spectrograms, pitch estimation).
* **NumPy** – For numerical computations.
* **Soundfile (sf)** – For audio I/O operations.
* **Torchaudio** – For resampling and additional audio transformations.
* **Pyannote.audio** – Provides a pre-trained speaker diarization pipeline.
* **SpeechBrain** – Offers the ECAPA-TDNN model for extracting speaker embeddings.
* **Transformers (Wav2Vec2)** – Although imported, the current code primarily leverages other components.
* **Warnings** – To suppress non-critical warning messages.

**2. Architecture Overview**

The overall system follows a modular design where each component is responsible for a specific task. The high-level architecture is as follows:

1. **Audio Preprocessing Module:**
   * Loads and resamples audio.
   * Extracts normalized mel spectrogram features for use in further classification.
2. **Emotion Classifier Module (CNN + LSTM):**
   * Uses convolutional layers to extract local spectral–temporal features from the mel spectrogram.
   * Applies average pooling across the frequency dimension.
   * Uses a bidirectional LSTM to capture temporal dynamics.
   * Uses global average pooling over time followed by a fully connected layer to output emotion logits.
3. **Gender Classifier Module (Using ECAPA-TDNN):**
   * Uses the pre-trained ECAPA-TDNN model (from SpeechBrain) to extract speaker embeddings.
   * Applies statistical analysis (comparing high- vs. low-frequency components of the embedding) and reinforces this with pitch-based evidence to decide the gender.
4. **Speech Diarization Module:**
   * Uses the pre-trained pyannote.audio pipeline to segment the audio into different speaker turns.
   * Each segment is then analyzed for emotion and gender.
5. **Audio Analyzer Module:**
   * Combines preprocessing, diarization, emotion classification, and gender classification.
   * Iterates over diarized segments, applies both classifiers, and aggregates results.

**3. Detailed Component Breakdown**

**3.1 Audio Preprocessing**

* **Functionality:**
  + **Audio Loading:** Uses Librosa to load the audio file at a fixed sample rate (default 16 kHz) and ensures the audio isn’t empty.
  + **Feature Extraction:**
    - Extracts a mel spectrogram (using Librosa’s melspectrogram function).
    - Converts power to decibels (dB) for better representation.
    - Normalizes the spectrogram (zero mean and unit variance).
    - Converts the normalized spectrogram to a PyTorch tensor with shape (1, 1, n\_mels, time) suitable for CNN input.
* **Techniques:**
  + Signal resampling and normalization.
  + Spectral analysis using the mel scale.

**3.2 Emotion Classification (CNN + LSTM)**

* **Architecture:**
  + **Convolutional Layers:**
    - First layer: 2D convolution with 16 filters, batch normalization, ReLU activation, followed by a max pooling over both frequency and time.
    - Second layer: 2D convolution with an increased number of channels (32 by default), followed by batch normalization, ReLU, and further pooling (with kernel size that preserves more temporal resolution).
  + **Temporal Modeling (LSTM):**
    - A bidirectional LSTM captures temporal dependencies across frames.
    - Global average pooling over the LSTM outputs generates a fixed-length feature vector.
  + **Classification:**
    - A fully connected layer maps the pooled LSTM output to the number of emotion classes (e.g., neutral, happy, sad, angry).
* **Algorithms and Techniques:**
  + **Convolutional Neural Networks (CNN):** Extracts local features from time-frequency representations.
  + **Recurrent Neural Networks (LSTM):** Captures sequential dependencies in the temporal domain.
  + **Global Average Pooling:** Reduces variable-length sequences to fixed-size representations.
  + **Activation Functions and Batch Normalization:** Improve training stability and convergence.

**3.3 Gender Classification (Using ECAPA-TDNN from SpeechBrain)**

* **Pre-trained Model:**
  + **ECAPA-TDNN:**
    - A variant of the Time Delay Neural Network optimized for speaker recognition.
    - It extracts robust speaker embeddings which encapsulate characteristics like pitch and timbre.
    - Pre-trained on large-scale datasets (e.g., VoxCeleb) to capture speaker-specific traits.
* **Algorithm:**
  + **Embedding Extraction:**
    - The audio segment is converted to a tensor, resampled if necessary, and passed to the ECAPA-TDNN model to extract speaker embeddings.
  + **Statistical Analysis of Embeddings:**
    - The embedding vector is split into two halves. The idea is that the higher frequency half tends to be more prominent in female voices.
    - Computes the mean of the higher and lower halves of the embedding vector.
  + **Pitch Extraction as Auxiliary Evidence:**
    - Uses Librosa’s piptrack to extract pitch information from the waveform.
    - Computes a median pitch value; higher median pitches suggest a female voice.
  + **Decision Making:**
    - A gender score is calculated by combining the difference between high and low frequency components of the embedding and the pitch evidence.
    - A positive score indicates “female” and a non-positive score indicates “male.”
* **Techniques:**
  + **Deep Speaker Embedding:** Using a pre-trained deep model (ECAPA-TDNN) to capture voice characteristics.
  + **Heuristic Statistical Analysis:** Simple thresholding and averaging are used to infer gender.
  + **Pitch Analysis:** Extracts additional prosodic features to support the embedding-based prediction.

**3.4 Speech Diarization**

* **Functionality:**
  + Uses the pre-trained pyannote.audio speaker diarization pipeline.
  + **Process:**
    - The pipeline takes an audio file and automatically detects speaker boundaries.
    - It outputs segments with associated speaker labels, which are then used for per-segment emotion and gender analysis.
* **Underlying Algorithms (Pre-trained):**
  + **Speaker Embedding Extraction:** Similar to the gender classifier, embeddings capture speaker-specific features.
  + **Clustering:** Likely involves techniques such as Agglomerative Clustering or spectral clustering to group segments by speaker identity.
  + **Voice Activity Detection (VAD):** To identify active speech regions before segmentation.
* **Techniques:**
  + **Deep Learning for Diarization:** Combines neural network-based feature extraction with clustering methods.
  + **Pre-trained Models:** The diarization pipeline is provided as a ready-to-use component from pyannote.audio.

**3.5 Audio Analyzer (Main Pipeline)**

* **Workflow:**
  1. **Audio Loading and Preprocessing:**
     + The audio file is loaded, resampled, and its mel spectrogram is computed.
  2. **Speaker Diarization:**
     + The diarization pipeline segments the audio into speaker-specific intervals.
  3. **Segment Analysis:**
     + For each diarized segment:
       - The emotion classifier predicts the emotion from the mel spectrogram.
       - The gender classifier predicts the gender using speaker embeddings and pitch analysis.
  4. **Aggregation:**
     + The results for each segment are aggregated by speaker to compute overall statistics (e.g., most common emotion, gender confidence).
* **Techniques and Workflow:**
  1. **Modular Processing:** Each segment is independently analyzed.
  2. **Aggregation:** Final results are summarized per speaker based on frequency counts.
  3. **Error Handling:** Robust try/except blocks ensure that processing errors in one segment do not halt the entire pipeline.

**4. Pre-trained Models: Algorithms & Techniques**

**4.1 ECAPA-TDNN (For Gender Classification)**

* **Architecture:**
  + **Time Delay Neural Network (TDNN):** Processes sequential data by using temporal context.
  + **Enhanced Channel Attention:** Incorporates channel-wise attention mechanisms to focus on informative features.
  + **Multi-scale Feature Extraction:** Often uses multiple kernel sizes or residual connections to capture varying temporal resolutions.
  + **Training:** Typically trained on large-scale speaker recognition datasets (e.g., VoxCeleb) to learn discriminative speaker embeddings.
* **Usage in the Project:**
  + Extracts speaker embeddings from each audio segment.
  + The embedding is then statistically analyzed (splitting into halves and comparing averages) to infer gender.

**4.2 Pyannote.audio’s Speaker Diarization Pipeline**

* **Architecture and Techniques:**
  + **Voice Activity Detection (VAD):** First, regions of active speech are detected.
  + **Speaker Embedding Extraction:** Uses a deep neural network (often similar to or based on models like ECAPA-TDNN) to extract embeddings from speech segments.
  + **Clustering:** Uses clustering techniques (e.g., Agglomerative Clustering) to group embeddings corresponding to the same speaker.
  + **Post-processing:** Refines segment boundaries and labels to generate a coherent diarization output.
* **Usage in the Project:**
  + The diarization pipeline automatically processes the entire audio file, returning time-stamped segments with speaker labels that are then used to guide the subsequent emotion and gender classification tasks.