**Audio Analysis Documentation**

I'll provide comprehensive documentation for your audio analysis application, including flowcharts, architecture diagrams, and algorithm explanations that would be suitable for a research paper.

**System Overview**

The Audio Analysis App is a comprehensive audio processing pipeline designed to extract meaningful information from speech recordings. It integrates multiple deep learning models to perform:

1. Speaker diarization (identifying who speaks when)
2. Gender detection for each speaker
3. Emotion recognition for each speech segment

**Architecture Diagram**

Audio Analysis System Architecture

Image

**Workflow Flowchart**

Audio Analysis Workflow

Image

**Component Details and Algorithms**

**1. Audio Preprocessing Module**

The AudioPreprocessor class handles loading audio files and extracting features suitable for downstream models.

**Key functions:**

* Loading audio with librosa
* Resampling to a fixed sample rate (16kHz)
* Mel spectrogram extraction
* Feature normalization

**Algorithm Details:**

1. Load audio using librosa, resampling to 16kHz
2. Extract mel spectrogram features with 128 mel filters and frequency max of 8kHz
3. Convert power spectrogram to decibels
4. Normalize features using mean and standard deviation
5. Create a tensor with proper dimensions for model input

**2. Speaker Diarization Module**

The application uses the PyAnnote Speaker Diarization pipeline to segment audio by speaker.

**Model Details:**

* **Model**: pyannote/speaker-diarization
* **Architecture**: Neural diarization system based on end-to-end neural speaker segmentation
* **Implementation**: Uses the Pipeline API from pyannote.audio

**Algorithm Flow:**

1. Process audio through self-attentive speaker embedding network
2. Generate frame-level speaker embeddings
3. Perform temporal clustering to identify speaker segments
4. Apply overlap detection to handle multi-speaker segments
5. Return timestamped segments with speaker identities

**3. Gender Classification Module**

The GenderClassifierECAPATDNN class utilizes an ECAPA-TDNN based model to classify speaker gender.

**Model Details:**

* **Model**: JaesungHuh/voice-gender-classifier
* **Architecture**: ECAPA-TDNN (Emphasized Channel Attention, Propagation and Aggregation in Time Delay Neural Network)
* **Features**: Leverages temporal context and channel attention mechanisms

**Algorithm Flow:**

1. Extract speaker-specific segments from diarization
2. Process audio through ECAPA-TDNN for speaker embedding extraction
3. Apply gender classification head to embeddings
4. Output binary gender classification (male/female)
5. Include fallback acoustic-based method if model fails

**4. Emotion Detection Module**

The HatmanEmotionDetector class implements a Wav2Vec2-based emotion recognition model.

**Model Details:**

* **Model**: Hatman/audio-emotion-detection
* **Architecture**: Fine-tuned Wav2Vec2 transformer model
* **Classes**: Classifies emotions into angry, happy, sad, neutral, fear, disgust, and surprise

**Detailed Algorithm Explanations**

**Wav2Vec2 Emotion Detection Algorithm**

The Wav2Vec2 emotion detection model uses a self-supervised learning approach to understand audio representations, which is then fine-tuned for emotion classification:

1. **Feature Extraction**:
   * The raw audio waveform is processed through a multi-layer CNN encoder
   * 7 convolutional blocks with GELU activations extract feature representations
   * These features capture phonetic and prosodic information relevant to emotion
2. **Contextual Representation**:
   * The extracted features are passed through a 12-layer transformer encoder
   * Each transformer layer uses multi-head self-attention to capture contextual dependencies
   * This allows the model to understand emotional context across time
3. **Quantization Module**:
   * During pre-training, the model learned to predict quantized representations
   * This quantization process helps in learning discrete speech units
   * It forces the model to focus on the most important aspects of the signal
4. **Fine-tuning for Emotion Classification**:
   * The pre-trained model is adapted for emotion detection through fine-tuning
   * A classification head is added on top of the transformer outputs
   * The model is trained on emotion-labeled speech data
   * A weighted average of the transformer outputs is used for final prediction
5. **Inference Process**:
   * Audio is segmented into fixed-length chunks
   * Each chunk is processed through the model independently
   * The classification head outputs probabilities for each emotion class
   * The emotion with the highest probability is selected as the prediction

**PyAnnote Speaker Diarization Algorithm**

The PyAnnote speaker diarization pipeline performs the following steps:

1. **Voice Activity Detection (VAD)**:
   * Identifies regions containing speech vs. non-speech
   * Uses a convolutional neural network trained on diverse audio datasets
   * Outputs binary speech/non-speech segmentation with timestamps
2. **Feature Extraction**:
   * Extracts MFCC or mel-spectrogram features from the audio
   * Applies normalization and augmentation techniques
   * Prepares features for the speaker embedding model
3. **Speaker Embedding**:
   * Utilizes an ECAPA-TDNN architecture to extract speaker embeddings
   * Processes sliding windows of audio (typically 1.5-2 seconds)
   * Generates 192-dimensional vectors that represent speaker characteristics
   * These embeddings cluster together for the same speaker across different utterances
4. **Speaker Change Detection**:
   * Analyzes the similarity between consecutive embeddings
   * Detects points where the speaker likely changes
   * Uses a threshold-based approach or neural change detector
5. **Clustering**:
   * Applies agglomerative hierarchical clustering to speaker embeddings
   * Merges similar segments based on cosine similarity
   * Determines optimal number of clusters (speakers) automatically
   * Each cluster corresponds to a unique speaker
6. **Overlap Detection**:
   * Identifies regions where multiple speakers talk simultaneously
   * Uses a dedicated model to detect overlapped speech
   * Enables attribution of speech to multiple speakers in these regions
7. **Resegmentation**:
   * Refines initial diarization results using Hidden Markov Models
   * Applies Viterbi decoding to find optimal speaker assignments
   * Corrects potential errors in the initial clustering
   * Ensures smooth and consistent speaker transitions
8. **Post-processing**:
   * Removes very short segments (typically less than 0.5 seconds)
   * Merges adjacent segments from the same speaker
   * Applies temporal constraints to avoid rapid speaker switching
   * Produces the final diarized output with speaker labels and timestamps

**ECAPA-TDNN Gender Classification Algorithm**

The ECAPA-TDNN (Emphasized Channel Attention, Propagation and Aggregation in Time Delay Neural Network) for gender classification:

1. **Input Processing**:
   * Audio segments are converted to 16kHz mono
   * Features are extracted as 80-dimensional mel-spectrograms
   * Normalization is applied for consistent processing
2. **Frame-level Feature Extraction**:
   * The first convolutional layer processes the raw features
   * Applies temporal convolutions with kernel size 5
   * Extracts basic acoustic features relevant to gender characteristics
3. **SE-Res2Block Processing**:
   * Multiple SE-Res2Blocks process the features
   * Each block has scale-invariant feature extraction
   * Channel attention recalibrates feature importance
   * Residual connections preserve information flow
4. **Attentive Statistical Pooling**:
   * Learns to assign importance weights to different time frames
   * Computes weighted mean and standard deviation
   * Aggregates temporal information into fixed-dimensional representations
   * Creates utterance-level embeddings
5. **Classification Head**:
   * Takes the pooled embeddings as input
   * Contains fully-connected layers with batch normalization
   * Binary output for gender classification (male/female)
   * Uses softmax activation for final probability
6. **Fallback Mechanism**:
   * In case of model prediction failure, the system includes acoustic-based fallbacks
   * Analyzes pitch and formant frequencies
   * Uses pitch median values (typically higher for females)
   * Ensures robustness in various acoustic conditions

**Integration Module**

The AudioAnalyzer class integrates all these models into a cohesive pipeline:

1. Loads and preprocesses the input audio
2. Performs speaker diarization to segment the audio by speaker
3. For each segment:
   * Extracts the audio for that segment
   * Performs gender classification using the ECAPA-TDNN model
   * Performs emotion detection using the Wav2Vec2-based model
4. Combines the results into a unified timeline
5. Generates visualizations and statistical summaries