Simple Audio Classification Using Feature Extraction

Part 1

Objective:

Classify audio signals based on signal processing techniques and design filters to improve classification performance. You will be provided with three folders:

- `class_1`
- `class_2`
- `unknown`

Each folder contains `.wav` audio files. The `unknown` folder contains samples from either `class_1` or `class_2`. Your tasks are to:

1. Classify the unknown audio files correctly.
2. Be able to classify a random audio file during live evaluation

---

Instructions:

1. Feature Extraction:

You must extract features from all audio files using one or more of the following methods:

- **MFCC (Mel-Frequency Cepstral Coefficients)**
- **FFT (Fast Fourier Transform)**

Choose whichever is more appropriate or try both and compare the results.

2. Similarity/Distance Measures:

To match files from the `unknown` folder with either `class_1` or `class_2`, compare feature vectors using at least one of the following:

- **Cosine Similarity**
- **Euclidean Distance**
- **Manhattan Distance**

You are free to test multiple methods and discuss their performance in your report.

---

3. Classification Task:

- Extract features from all files in `class_1` and `class_2`.
- For each file in the `unknown` folder, compare it with files in `class_1` and `class_2` using your chosen similarity/distance metric.
- Assign the file to the class with the closest match (minimum distance or maximum similarity).

Part 2

Filter Design

Objective

You are provided with a dataset located in the folder named `filter`, which contains audio recordings from two classes: **ambulance** and **firetruck**. Your task is to design a digital signal processing (DSP) pipeline that can accurately distinguish between these two types of emergency vehicle sounds.

Using the training set, analyze the frequency content of each class and design a filter-based classification approach. Then, apply your method to the test set and calculate the classification accuracy. Include your final classification performance in your report.

Instructions:

1. **Explore and Analyze the Spectrum**

    - Load and inspect the audio files in the training set.
    - Identify distinguishing frequency bands between the ambulance and firetruck signals.

2. **Design Digital Filters**

    - Based on your spectral analysis, design **two or more bandpass filters** targeting frequency regions that best isolate the differences between the two classes.

3. **Compute Filtered Energy Ratios**

    - For each audio file, filter the signal using the designed filters.
    - Calculate energy ratios (e.g., energy in Filter A / energy in Filter B) as features for classification.

4. **Set Classification Thresholds**

    - Analyze the distribution of energy ratios in the training data.
    - Set a decision threshold (or multiple thresholds) to separate the two classes based on these features.

5. **Test and Evaluate**

    - Apply the same filtering and classification logic to the audio files in the test set.
    - Measure the classification **accuracy** (percentage of correctly classified samples).
    - Include this accuracy in your final report.

Report Guidelines:

Each team must prepare a **detailed report** in PDF format. Append both Part 1 and Part2 process in the same report. Your report should include the following:

A. Introduction

- Brief explanation of the problem and your approach.

B. Methodology

- Description of the feature extraction method(s) used.
- Explanation of the distance/similarity metrics chosen.
- Justification of why you chose certain methods.
- Explain how the filter was created (type, order, tool used)
- Justify why you selected that filter type and frequency range.
- Describe how you set the threshold for filtering

C. Implementation Details

- Overview of your code structure.
- Description of how the classification is done (For part 1 and part 2).
- Tools used (only **MATLAB** or **C++** are allowed).

D. Results

- Classification results for each file in `unknown` in a table.
- Accuracy or confidence in your classification.(For part 1 and part 2)
- No visualization of signal is required.

E. Conclusion

- Summary of what worked well and what challenges you faced.
- Any improvements or alternative approaches you considered.

Evaluation Procedure:

- **In-Class Evaluation:** During the evaluation session, you will be asked to explain your code and methodology.
- **Live Classification:** You will be given a **random audio file** and asked to classify it on the spot using your code.
- You **must use the same code** submitted in Teams. No modifications are allowed during the evaluation.