Signal Processing Lab Project - 2023

Part 1 - Echo creation

In signal processing terms, an "echo" refers to a delayed and attenuated version of a sound or signal that is heard after the original sound. It is a time-delayed replication of the original signal, where the delayed version is often quieter than the original due to the signal being reflected off surfaces or travelling a longer path. Echoes are commonly encountered in various acoustic environments, such as rooms, hallways, and outdoor spaces.

In this section, you will explore the world of audio signal processing by creating an echo effect for a given audio file.

Task :

- Input Audio File:
  - You will be provided with an input audio file in a specific format (e.g., WAV or MP3).
- Create an Echo Effect:
  - Your task is to apply signal processing techniques to create an echo effect for the given audio.
  - The echo effect should be audible but not overpowering, simulating a natural echo found in acoustic environments.
- Adjustable Parameters:
  - You can choose the delay time (in seconds) for the echo.
- Output Audio File:
  - You should play the resulting audio, which includes the echo effect.

Part 2 - Cancel the echo

In this section, you will be presented with an audio signal containing an echo with non-uniform delays. Your task is to implement an echo cancellation algorithm to remove the echo components from the signal, leaving only the original, clean audio.

Task :

- Input Audio Signal:
  - You will receive an audio signal in a specific format (e.g., WAV or MP3), which contains both the original audio and the echo with non-uniform delays.
- Echo Cancellation:
  - Your task is to design and implement an algorithm to cancel the echo in the signal.
  - The echo cancellation algorithm should effectively remove the echo components, leaving behind the original, clean audio.
- Algorithm Implementation:

    - You can use any signal processing techniques or algorithms you deem appropriate for this task. Experimentation and creativity are encouraged.
    - You might need to identify and model the non-uniform delays in the signal.
- Output Clean Audio Signal:
  Play the resulting audio, which should ideally contain only the original, unechoed sound.

Part 3: What is this noise?

In this section, you will work on the task of classifying background noise in music recordings without removing the noise. Your objective is to accurately identify and categorise the type of noise present in the recording, distinguishing source of origin.

The types of Noises can be:

- Fan
- Pressure cooker
- Water Pump
- Traffic

Task :

- Input Music Recording:
    - You will be provided with a music recording in a specific audio format (e.g., WAV or MP3), which contains both the original music and background noise from indoor and outdoor sources.
- Background Noise Classification:
    - Develop an algorithm to classify the background noise sources. Your goal is to accurately identify and categorize the type of noise (e.g., fan, pressure cooker/mixer, water pump, traffic).
    - Your algorithm should indicate the presence of specific noise sources and label them accordingly.
- Algorithm Implementation:
    - You can use signal processing to achieve your classification task. Experimentation and creativity are encouraged.
- Output Noise Classification:
    - Provide a detailed classification report that specifies the types of noises in the audio recording.

General Instructions:

1. You will be given 1 github repository per team, please make sure to submit all the codes and outputs in the repo itself.
2. Structure of the repo;

    - Team_Name(Main folder):
        - Input Audio
        - Src (all the codes are supposed to be here)
            - q1
            - q2
            - q3
        - Output Audio/Plots if any
        - Report
3. You are required to submit a detailed report that explains your approach towards the problem clearly. Follow the below pattern -
    - Objective/Aim
    - Input
    - Problem solving approach
    - Results
    - Conclusion
4. Programming Language - **MATLAB**
5. **Deadline for code submission** - 4th December 2023 11:55 PM
6. Sample input audio files will be provided on moodle
7. During Evaluation the test audio files will be different so please don't hardcore any function.
8. We expect everyone to write modular code, in short every question should follow the following pattern as given below -

```
% Function 1: Load the input audio file
function audio = loadAudioFile(filename)
    % Load the audio file
    audio = audioread(filename);
end

% Function 2: Noise classification
function noiseType = classifyNoise(audio)
    % Implement your noise classification algorithm here
    % You can use signal processing and machine learning techniques
    % to classify the type of noise (e.g., fan, pressure cooker, water pump,
traffic).

    % Replace the following line with your classification logic
    noiseType = 'Unknown';
end

% Main script
inputAudioFile = 'input_audio.wav';  % Replace with the actual filename

% Step 1: Load the input audio file
audio = loadAudioFile(inputAudioFile);

% Step 2: Classify the noise
noiseType = classifyNoise(audio);
```

```
% Display the classification result
disp(['The detected noise type is: ' noiseType]);

% Optionally, you can save the classification result to a file or return it.
```

Note: Function 1 and Function 2 are to be in 2 separate files.