German University in Cairo GUC
Faculty of Media Engineering and Technology
DMET 603—Digital Signal Processing
Spring Semester 2025

Prof. Dr. Mohammed Abdel-Megeed Salem
Eng. Yomna Islam Youssef Alayary

Assignment 2

Practical

A data scientist is building a lightweight music genre classifier using frequency characteristics of audio. Given a dataset of music clips (each 30–60 seconds), the goal is to classify songs into genres based on the dominant frequency component derived from the signal's Discrete Fourier Transform (DFT). The twist: instead of using full spectrograms or multiple features, the model will rely on a single frequency-based feature extracted manually using DFT computed from scratch.

Tasks:

1. **Load the Data**:
    - Depending on your assigned group, you will be provided with a subset of the GTZAN Genre Collection on **CMS**: **Group A or Group B** will each receive 15 audio files (5-30 seconds each).
    - You can download the audio files using: https://drive.google.com/drive/folders/11gp11jek1E1z_8jZndb3hLH6zest1YI2?usp=sharing
    - Genres in classification are: **Classical**, **Disco**, and **Rock**
    - Ensure all files are **mono** and loaded with the **same sampling rate** (e.g., 3000 Hz) for consistency.
    - Optionally, trim or pad audio clips to a fixed length for uniform analysis.

2. **Compute Discrete Fourier Transform (DFT) From Scratch**:
    - Implement the DFT manually using its mathematical definition **from scratch** to convert to the frequency domain. Avoid using np.fft or other fast libraries
    - Using any built-in functions for DFT computation **will NOT be accepted.**

3. **Extract the Dominant Frequency:**
    - From the DFT output, find the frequency bin with the maximum magnitude.
    - Use this single dominant frequency as the representative feature of the song.

4. **Cluster Genre Patterns Using K-Means:**
    - Use the dataset you created above where each sample contains the **dominant frequency** extracted from an audio clip.
    - Apply **K-Means clustering** with k=3 to group songs based on frequency patterns into 3 clusters.

- The three centers of clusters used in k-means should be decided depending on the frequencies of the genre they represent.

- Use the standard K-Means algorithm to minimize the within-cluster sum of squares:

$$J = \sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - \mu_i\|^2$$

  , where μ_i = the dominant frequency of the center
- After clustering, assign each song to a cluster (e.g., Classical, Disco, Rock), depending on which cluster is the closest to the song.

5. **Visualization and Analysis**:
   - **Printed Array of Audio File Details:** Print an array or table displaying each audio file's title, its corresponding dominant frequency (in Hz), and the genre it was classified as.
   - **Bar Graph of Genre Classifications:** Plot a bar graph showing how many audio signals were classified into each genre.
   - **Frequency Ranges by Genre:** For each genre, plot the range of dominant frequencies that the classifier associates with it. This can be done by collecting the dominant frequencies from all samples of that genre and showing the minimum and maximum values in a bar chart or box plot.

Regulations:

1. You may work in teams of up to **4 students,** and the teams will not change for the rest of the semester. The deadline for submitting teams is **Saturday 12/4/2025.** You can submit your teams here: https://forms.gle/eqbXwHwXcdKbzqYPA
2. The deadline is **Friday 25/4/2025 at 11:59 pm**.
3. Submit your work to dlcv.guc@gmail.com.
4. A deduction of 5% for the late submissions till 5:59 am of the next morning.
5. The Subject of the Email is **DSP_Assign2_[ID1_ID2_ID3_ID4]**
6. The deliverables are (in one zip file named as the email subject, rar files are not accepted): One .py code of practical part (you can share Google Colab link directly).