SGN-16006 Bachelors Laboratory Course in Signal Processing Lab02 - Computational Auditory Scene Recognition: Classification of Environmental Sounds with Python

Hung Nguyen - 272585
Khoa Nguyen - 272580

***Abstract*— In this report, we present an implementation and results for Computational Auditory Scene Recognition using Sub-band energy ratio for Framewise feature extraction and k-Nearest Neighbors to classify the new Auditory Scene data.**

I. INTRODUCTION

Computational auditory scene recognition refers to a process of recognizing the location of the device (office, a street, or a train...) based on the characteristics of the audio signal recorded by that device. In this exercise, we follow the idea presented in [1] to implement a Python program that predicts the recording environment from an audio clip.

II. FRAMEWISE FEATURE EXTRACTION: SUB-BAND ENERGY RATIO

Audio signal processing, for example audio signal analysis and classification typically relies on framewise processing. Framewise processing means that the signal is divided into short segments called frames. Typically, a smooth window such as Hanning window is used to multiply the signal values in each frame.
We try this method on the audio file 'signal.wav' which has the sampling frequency of 8000Hz. Then we divide the audio clip into frames by using Hanning window and frame length of 30 ms and 15 ms overlap of adjacent frames. One frame contains 240 samples.

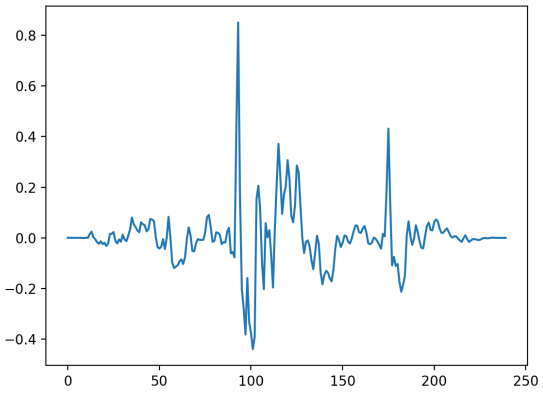

Fig. 1. The waveform of the 50th Hanning-windowed frame

After that, we apply Discrete Fourier Transform (DFT) on each frame.

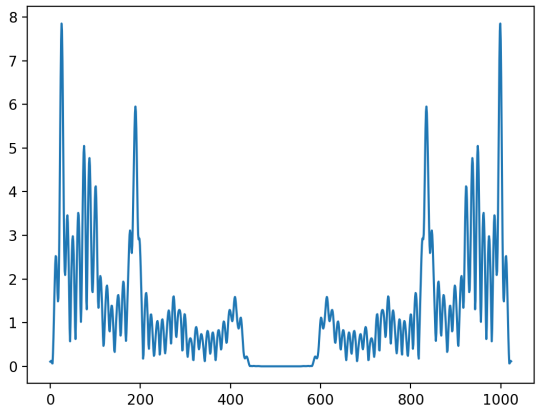

Fig. 2. The amplitude spectrum of the 50th signal frame

The sub-band energy ratio describes the relative energy on certain frequency bands. For each signal frame,we will extract a feature vector $x = [x(1), x(2), x(3), x(4)]^T$ containing the relative energy on the frequency bands 0-0.5 kHz, 0.5-1 kHz, 1-2 kHz and 2-4 kHz.
If the discrete Fourier transform (DFT) of a signal frame is denoted by $S = [S(1), S(2), ..., S(L)]^T$, the sub-band energy ratio can be computed as:

$$x(i) = \frac{\sum_{l=b_i}^{e_i} |S(l)|^2}{\sum_{l=0}^{L/2} |S(l)|^2} \quad (1)$$

where L is the number of frequency bins in DFT and b_i and e_i denote the first and last DFT.
Applying this on 'signal.wav', the feature values for the 50th frame is plotted in figure 3. The indices of the DFT bins belonging to the frequency band 12 kHz is in the range 128:256.

III. DATA PRE-PROCESSING

The data consists of 12 audio samples recorded in different environments (bus, office, etc.), each one last for 5 minutes. As a pre-processing step, we segment the audio files into one-second clips without any overlap between the audio clips.
First, we read the audio file and then sort their names based on a-z order.

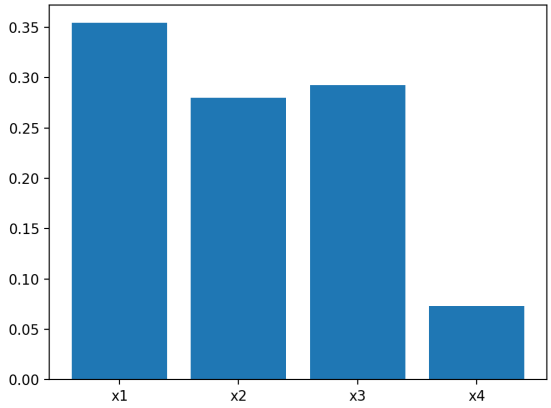

Fig. 3. The feature values for the 50th frame

After that, we process each training data clip framewise and extract the sub-band energy ratio values for each frame, then compute the average feature vectors.
The training data consist of 80% of the data, the rest is for testing. We split them using $sklearn.model\_selection.train\_test\_split$.

IV. K-NN CLASSIFICATION

K-NN is very simple and easy to implement. It only takes a few line of code and the whole algorithm is written in the function 'k-nn'.
We test our algorithm with 3 cases: 1-nn, 5-nn and 10-nn, the accuracies are 69.7%, 73.05% and 71.9% respectively. We can see that 5-nn is the best among the three of them.

```
[[75  0  0  0  0  0  1  0  0  1  0  0]
 [ 0 38  5 11  0  0  4  0 11  0  1  0]
 [ 0  9 48  4  0  0  2  0  0  0  0  0]
 [ 0  4  2 39  0  0  2  0  0  0  0  0]
 [ 0  3  0  1 47  0  5  1  7  0  1  2]
 [ 0  0  0  0  0 61  0  0  0  0  0  0]
 [ 0  7  0  4  3  0 39  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 19  4  2  8 11]
 [ 0  5  0  0  4  0  1  8 35  1  7  1]
 [ 0  0  0  0  0  0  0  6  3 33  5  4]
 [ 0  1  0  0  9  0  0 14  9  5 29  5]
 [ 0  0  0  0  1  0  0 10  2  0  1 39]]
```

Fig. 4. The confusion matrix for 1-nn

Looking at the confution matrices, we can see that the 2nd audio signal (bus1.wav), the 8th signal (shoppingcenter1.wav) and supermarket1.wav are hard to recognize because there are a lot of wrong predictions.

```
[[76  0  0  0  0  0  0  0  0  1  0  0]
 [ 0 39  7 10  1  0  2  0 10  0  1  0]
 [ 0  7 53  3  0  0  0  0  0  0  0  0]
 [ 0  0  3 42  0  0  2  0  0  0  0  0]
 [ 0  2  0  0 53  0  2  1  5  0  2  2]
 [ 0  0  0  0  0 61  0  0  0  0  0  0]
 [ 0  3  2  8  2  0 36  0  2  0  0  0]
 [ 0  0  0  0  0  0  0 19  3  2  9 11]
 [ 1  6  0  1  2  0  0  7 35  2  6  2]
 [ 0  0  0  0  0  0  0  4  2 37  4  4]
 [ 0  1  0  0  2  0  0 16 11  5 32  5]
 [ 0  0  0  0  1  0  0  9  0  0  0 43]]
```

Fig. 5. The confusion matrix for 5-nn

```
[[75  0  0  0  0  0  2  0  0  0  0  0]
 [ 0 30  9 10  8  0  1  0 11  0  1  0]
 [ 0  4 52  6  0  0  1  0  0  0  0  0]
 [ 0  0  1 44  1  0  1  0  0  0  0  0]
 [ 0  5  0  0 52  0  1  0  4  0  4  1]
 [ 0  0  0  0  0 61  0  0  0  0  0  0]
 [ 0  4  3  9  3  0 33  0  1  0  0  0]
 [ 0  0  0  0  0  0  0 18  2  3  7 14]
 [ 1  7  0  2  2  0  0  7 35  1  5  2]
 [ 0  0  0  0  0  0  0  4  1 38  2  6]
 [ 0  1  0  0  0  0  0 11 15  4 36  5]
 [ 0  0  0  0  0  0  0  8  0  0  1 44]]
```

Fig. 6. The confusion matrix for 10-nn

V. CONCLUSIONS

The lab work is very interesting and practical in which we learned more about feature extraction for audio signals, then classify them using knn - one of the most popular methods in classical machine learning, it triggers my curiosity about using deep learning (which I think we need more data) and other methods on this task. The instruction is clear, although there can be some improvements e.g. a figure of overlapping frames, an example for feature extraction using Sub-band energy ratio.
We try to write the program so that each part of the implementation is wrapped by a function, therefore we only need to make some modifications to get new data and apply the algorithm to recognize them. This way, we believe that we can use the program for realistic applications.

REFERENCES

[1] Tuomi J. Klapuri A. Huopaniemi J. Peltonen, V. and T. Sorsa. Computational auditory scene recognition.