VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Brno University of Technology
Faculty of Information Technology

ISS - Signals and systems

Analyzing the similarity of sound signals

Nikita Smirnov, xsmirn02

16. December 2024

Contents

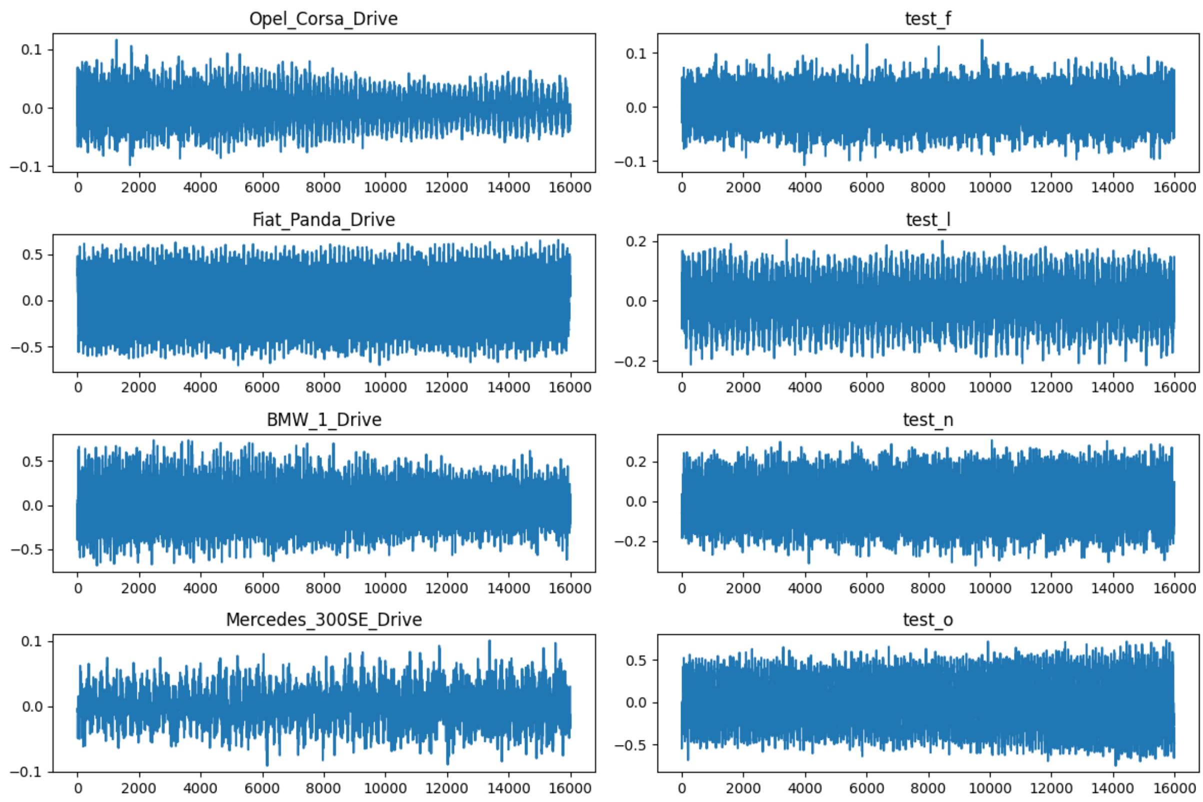
1 Introduction

This project focuses on identifying car models based on their engine sounds. We are provided with eight audio recordings: four reference recordings with known car models and four test recordings. The goal is to automatically match each test recording to its corresponding reference or determine if a test recording doesn't belong to any of the reference recordings.

2 Analysis and implementation details

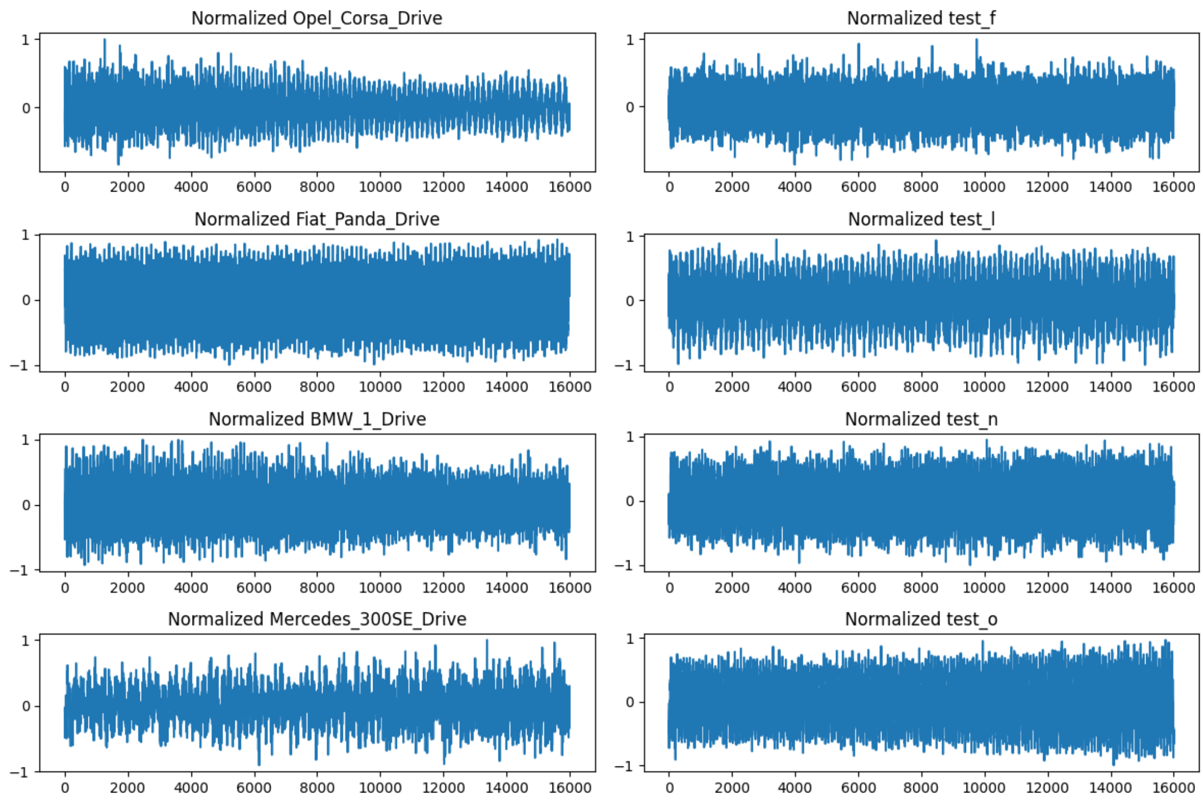
2.1 Initial analysis

In the manual analysis, after listening, no obvious correlations were found. After that, an attempt was made to analyze the amplitude of the signals, but even here, we do not see any obvious correlations.



2.2 Signal normalization

We can start by normalizing the signal by amplitude to allow easier analysis in the future, and we can also check again if there are obvious correlations between the signals. However, as we can see, there are no obvious correlations again.



2.2.1 The importance of normalization within this program

In the following, we will use MFCC, which is very robust to differences in the amplitude of the signals, as a metric for the similarity of the two signals, so this step can theoretically be skipped.
However, other metrics were used in the analysis process to calculate signal similarity, so signal normalization is still an important part of the project.

2.3 Calculating a metric for comparing signals

To compare the signals, we will use the Mel-Frequency Cepstral Coefficients as well as the Euclidean distance directly to calculate the "distance" between the signals.
This method was chosen among others because Mel-Frequency Cepstral Coefficients are ideal for analyzing the similarity of sound signals and have resistance to shifts like DTW. It is also robust to noise. These are key qualities for our project. However, of course, it is worth mentioning the computational complexity of this method.

2.3.1 MFCCs Calculation Steps

1. Framing and Windowing

   (a) The signal is divided into overlapping frames. The frame length is set to 25ms and the frame step is 10ms (calculated based on the sample rate of the audio signals). This allows for capturing both temporal and spectral information from the audio.

   (b) A Hamming window function is applied to each frame to reduce spectral leakage during the Fast Fourier Transform (FFT).

2. Fast Fourier Transform (FFT)

   (a) The FFT is applied to each windowed frame. This step transforms the signal from the time domain to the frequency domain, providing a representation of the signal's frequency components.

3. Mel-Frequency Cepstral Coefficients (MFCC) Calculation

   (a) Mel-Frequency transformation: The magnitude spectrum obtained from the FFT is converted using the magnitudes directly from FFT calculation

   (b) Cepstral coefficients: The logarithm of the magnitude spectrum is computed and the Inverse Fast Fourier Transform (IFFT) is applied. The first 13 cepstral coefficients (MFCCs) are extracted, as these coefficients are typically sufficient to represent the salient features of the speech or audio signal.

2.3.2 Metric for Distance Calculation

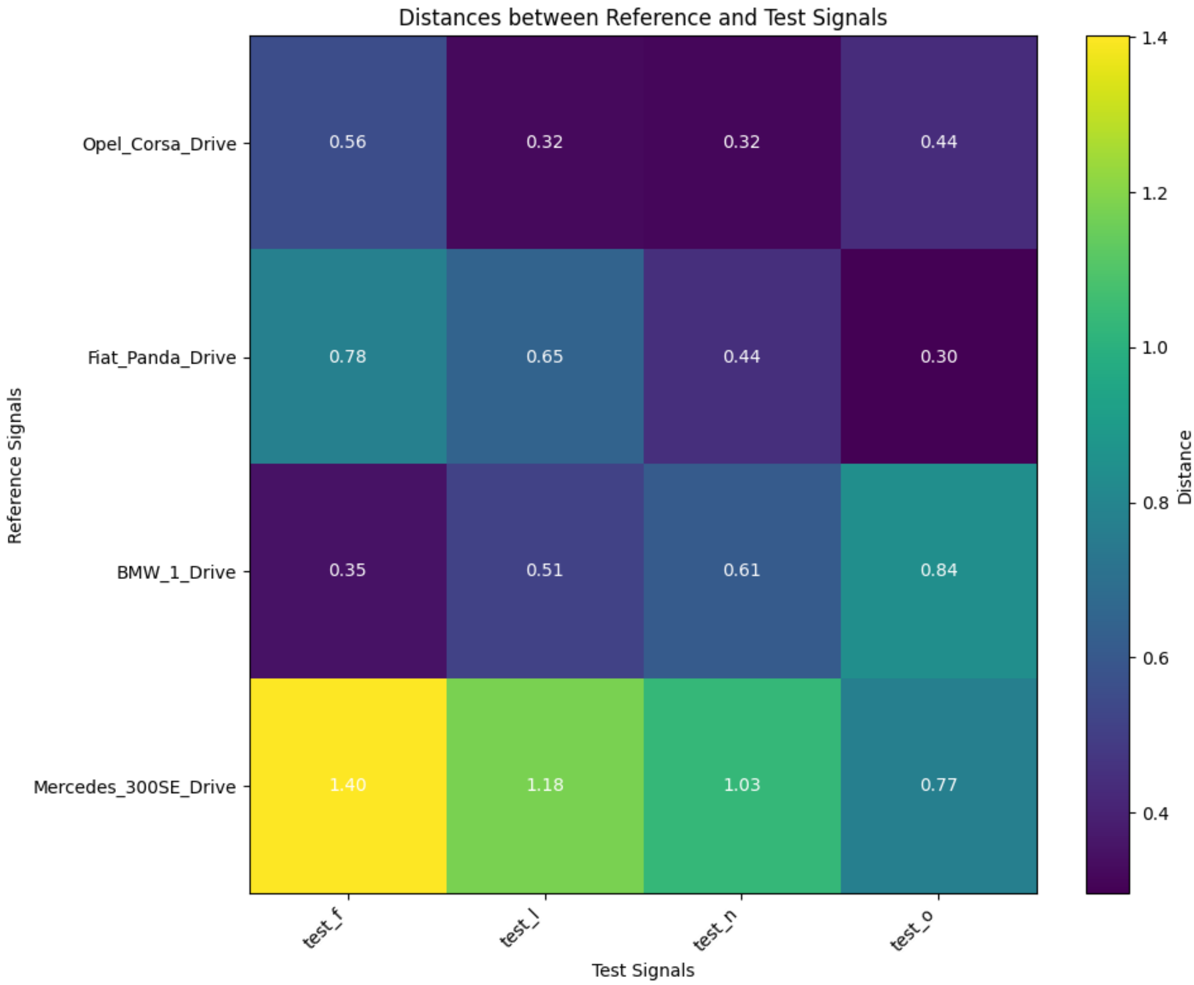
A metric is calculated to measure the similarity or dissimilarity between the MFCCs of reference signals and test signals. The function `calculate_metric` computes the Euclidean distance between MFCC vectors.

$$distance = \sqrt{\sum_{i=1}^{N} (ref\_mfcc[i] - test\_mfcc[i])^2} \tag{1}$$

where N is the number of MFCCs.

2.4 Similarity assessment based on the derived metrics.

At this point, we can use the previously described functions to calculate the similarity of the signals and draw conclusions regarding the belonging of the test signals.



As we can see, the analysis showed a clear correlation between test and example signals.

3 Results

Unfortunately, due to the specificity of reference signals, it is not possible to manually check the correctness of the analysis algorithm. However, our program shows this correlation between the signals, which on listening proved to be quite reliable:
$Signal Assignments:$
$test\_f -> BMW\_1\_Drive$
$test\_l -> Opel\_Corsa\_Drive$
$test\_n -> Opel\_Corsa\_Drive$
$test\_o -> Fiat\_Panda\_Drive$

It is worth mentioning again that these results are not fully reliable and are only an assumption based on the analysis performed.