Brno University of Technology
Faculty of Information Technology

Classification of Engine Sounds for Recognized Car Models

Project from subject:
Signály a systémy - ISS

Autor: Daniil Zverev (xzvere00)
Date: December 16, 2024

Contents

1 Overview

This project focuses on creating an automated system capable of categorizing engine sound recordings from various car models. Utilizing a dataset of known engine sounds from four distinct car models alongside four test recordings, the system aims to accurately assign each test sound to its corresponding model or categorize it as "Unknown" if it doesn't match any existing profiles. This classification process emulates human auditory recognition, offering a dependable and efficient approach for identifying sound sources.

2 Rationale Behind the Selected Approach

The primary goal was to develop a system that could automatically match unknown engine sounds to established car models or designate them as "Unknown." To accomplish this, a feature extraction and analysis strategy was employed, specifically leveraging Mel-Frequency Cepstral Coefficients (MFCC) alongside tailored spectral features. This combination is renowned in audio recognition domains for its resilience and precision in differentiating between various sound sources.

2.1 Fundamental Concepts and Techniques

2.1.1 Extracting Features

Mel-Frequency Cepstral Coefficients (MFCC): MFCCs serve as a fundamental feature set in the realms of speech and music recognition. They adeptly capture the tonal qualities of audio signals by simulating the human ear's frequency perception. The extraction process involves applying a Fourier transform to the audio signal, mapping the spectral power onto the mel scale, logging the values, and performing a discrete cosine transform. The resulting coefficients encapsulate the short-term power spectrum, making them effective for distinguishing between different engine noises.

Additional Spectral Features: Beyond MFCCs, several custom spectral features were derived to provide a more detailed frequency analysis:

- **Spectral Centroid:** Indicates the "center of mass" of the spectrum, reflecting where most frequency energy is concentrated.
- **Spectral Bandwidth:** Measures the width of the spectrum around the centroid, representing the signal's tonal complexity.
- **Spectral Rolloff:** Denotes the frequency below which a certain percentage (e.g., 85%) of the total spectral energy is contained, aiding in identifying the spread of higher frequencies.
- **Spectral Flatness:** Assesses how noise-like a signal is compared to being tone-like, assisting in differentiating between various sound textures.

2.1.2 Normalization and Standardization Processes

Normalization: To mitigate the impact of amplitude variations across different recordings, audio signals were normalized. This process scales the signal so that its peak absolute value is 1, ensuring consistency during feature extraction.

Standardization: Features extracted from both reference and test audio were standardized using the `StandardScaler` from `sklearn`. Standardization adjusts each feature to have a mean of zero and a standard deviation of one, which is essential for distance-based classification methods to operate without bias towards features with larger numerical ranges.

2.1.3 Distance Measurement

Euclidean Distance: The Euclidean distance metric was utilized to evaluate the similarity between feature vectors of test recordings and reference models. This metric calculates the straight-line distance between two points in the feature space, making it effective for identifying the closest reference match to each test recording based on their feature representations.

2.2 Justification for the Chosen Methodology

1. **Proven Effectiveness of MFCC:** MFCCs are well-established in capturing essential audio characteristics, making them suitable for distinguishing between different engine sounds.

2. **Comprehensive Spectral Analysis:** Incorporating additional spectral features alongside MFCCs provides a more nuanced representation of audio signals, enhancing classification accuracy.

3. **Implementation Simplicity:** Leveraging existing libraries such as `python_speech_features` and `scipy` facilitates efficient and reliable implementation without the need for developing complex algorithms from scratch.

4. **Robust Preprocessing Steps:** Normalization and standardization ensure that feature vectors are comparable across different recordings, eliminating biases caused by amplitude differences and varying feature scales.

3 Execution, Challenges, and Resolutions

3.1 Development Process

The system was developed using Python, incorporating libraries such as `numpy`, `scipy`, `matplotlib`, `soundfile`, and `python_speech_features`. The implementation involved the following key stages:

1. **Signal Loading and Preprocessing:**

    - **Importing WAV Files:** Audio data and sampling rates were read using the `soundfile` library.

- **Mono Conversion:** Stereo recordings were converted to mono by averaging the channels to ensure consistency in subsequent processing.
- **Signal Normalization:** Each audio signal was scaled to the range [-1, 1], standardizing amplitude levels across all recordings.

2. **Signal Visualization:**
   - **Waveform Plots:** Time-domain representations were generated using `matplotlib` to inspect signal amplitudes and identify any irregularities.
   - **Spectrogram Generation:** Spectrograms were created to examine the frequency content over time, providing a visual analysis of the signals' spectral properties.

3. **Feature Extraction:**
   - **MFCC Calculation:** The `mfcc` function from the `python_speech_features` library was used to extract MFCCs, capturing the key spectral characteristics of the audio.
   - **Spectral Properties Computation:** Additional spectral features such as centroid, bandwidth, rolloff, and flatness were derived using the Fast Fourier Transform (FFT) to provide a detailed frequency analysis.

4. **Feature Normalization:**
   - **Standardizing Features:** The `StandardScaler` from `sklearn` was applied to standardize all extracted features, ensuring equal contribution during distance calculations.

5. **Distance Computation and Classification:**
   - **Euclidean Distance Calculation:** Distances between each test feature vector and all reference feature vectors were computed to assess similarity.
   - **Distance Matrix and Histogram Creation:** A distance matrix was developed for visualizing distances, and a histogram was plotted to understand the distribution of these distances.
   - **Threshold Setting and Label Assignment:** A threshold, based on the median distance, was established to determine whether a test signal should be assigned to a known model or labeled as "Unknown."

3.2 Anticipated Challenges and Mitigation Strategies

1. **Handling Multi-Channel Audio:**
   - **Issue:** Stereo audio recordings introduce channel-specific variations that can complicate feature extraction.
   - **Solution:** Implemented a `convert_to_mono` function to average the stereo channels, ensuring a uniform single-channel input for feature processing.

2. **Dealing with Silent or Noisy Signals:**
    - **Issue:** Audio recordings with zero amplitude or excessive noise can lead to normalization errors, such as division by zero.
    - **Solution:** Added a conditional check in the `normalize_signal` function to bypass normalization if the maximum absolute value is zero, preventing computational errors.
3. **Calculating Spectral Properties for Low-Power Signals:**
    - **Issue:** Signals with negligible power can result in invalid spectral feature calculations, yielding misleading values.
    - **Solution:** Included a condition to set all spectral properties to zero when the total signal power is zero, maintaining the integrity of the feature vector.
4. **Determining the Optimal Classification Threshold:**
    - **Issue:** Selecting an inappropriate threshold can lead to incorrect classifications, either falsely identifying known models or mislabeling unknown ones.
    - **Solution:** Utilized the median of all Euclidean distances as the threshold value, providing a balanced approach to differentiate between known and unknown signals effectively.
5. **Ensuring Balanced Feature Scaling:**
    - **Issue:** Features with larger numerical ranges can disproportionately influence distance calculations, skewing classification outcomes.
    - **Solution:** Applied standardization using `StandardScaler` to normalize feature scales, ensuring equitable influence during distance measurements.
6. **Preventing Overfitting with Limited Reference Data:**
    - **Issue:** A small set of reference signals may cause the model to overfit, performing well on known data but poorly on new, unseen recordings.
    - **Solution:** Although not extensively addressed in this project due to dataset constraints, future enhancements could incorporate cross-validation techniques and expand the reference database to reduce overfitting risks.

4 Outcome and Findings

The developed system effectively categorized test recordings by matching them to known car models or designating them as "Unknown." The results were illustrated using a distance matrix and a histogram, which depicted the distribution of Euclidean distances between test and reference signals.

4.1 Primary Outcomes

4.1.1 Distance Matrix Visualization

The distance matrix showcased the Euclidean distances between each test recording and all reference recordings, facilitating the visual identification of the closest reference match for each test signal.

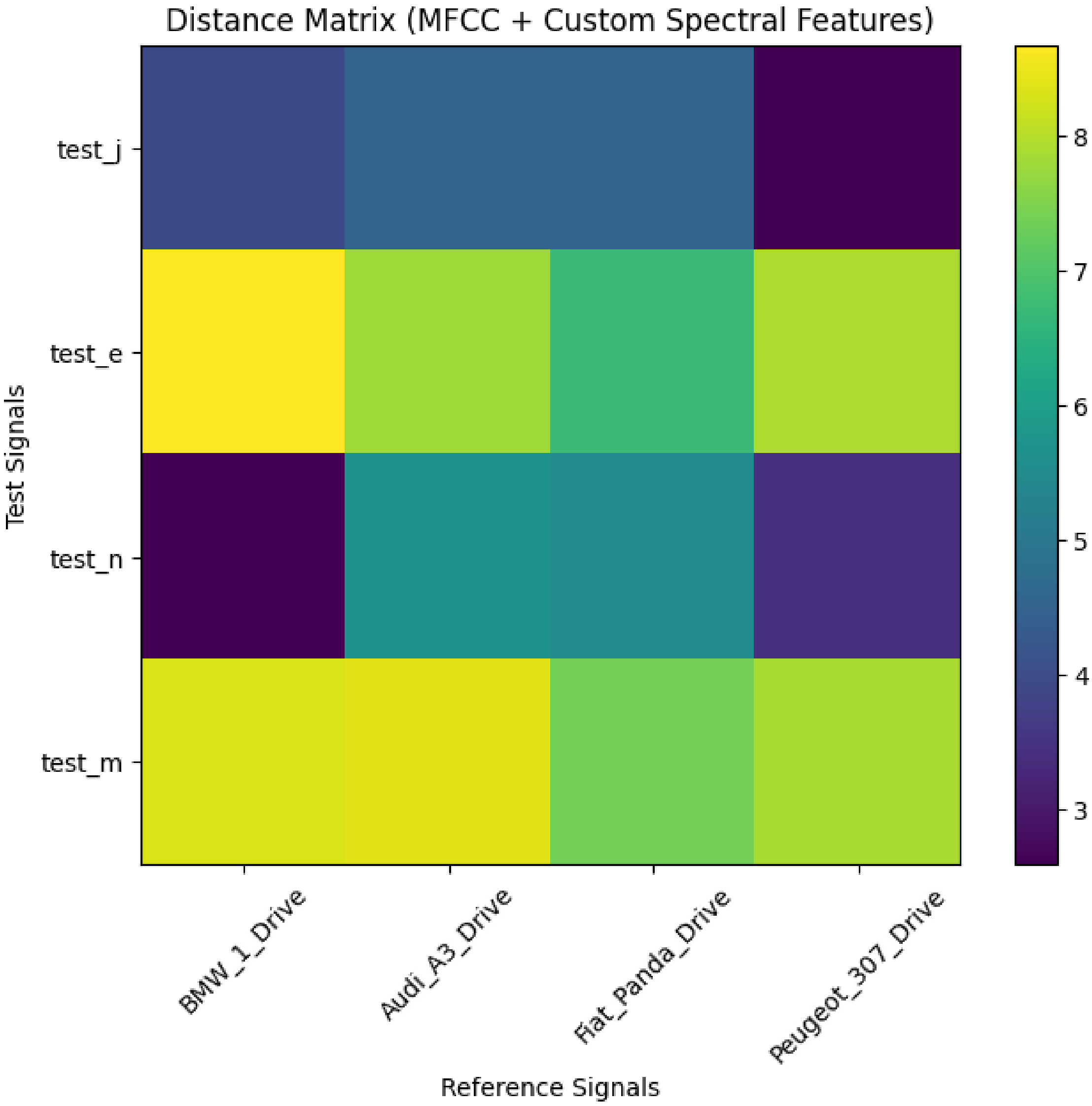

Figure 1: Euclidean Distance Matrix Combining MFCC and Custom Spectral Features

4.1.2 Distance Distribution Histogram

The histogram provided a comprehensive view of the distance distribution, assisting in the determination of an appropriate threshold for classification purposes.

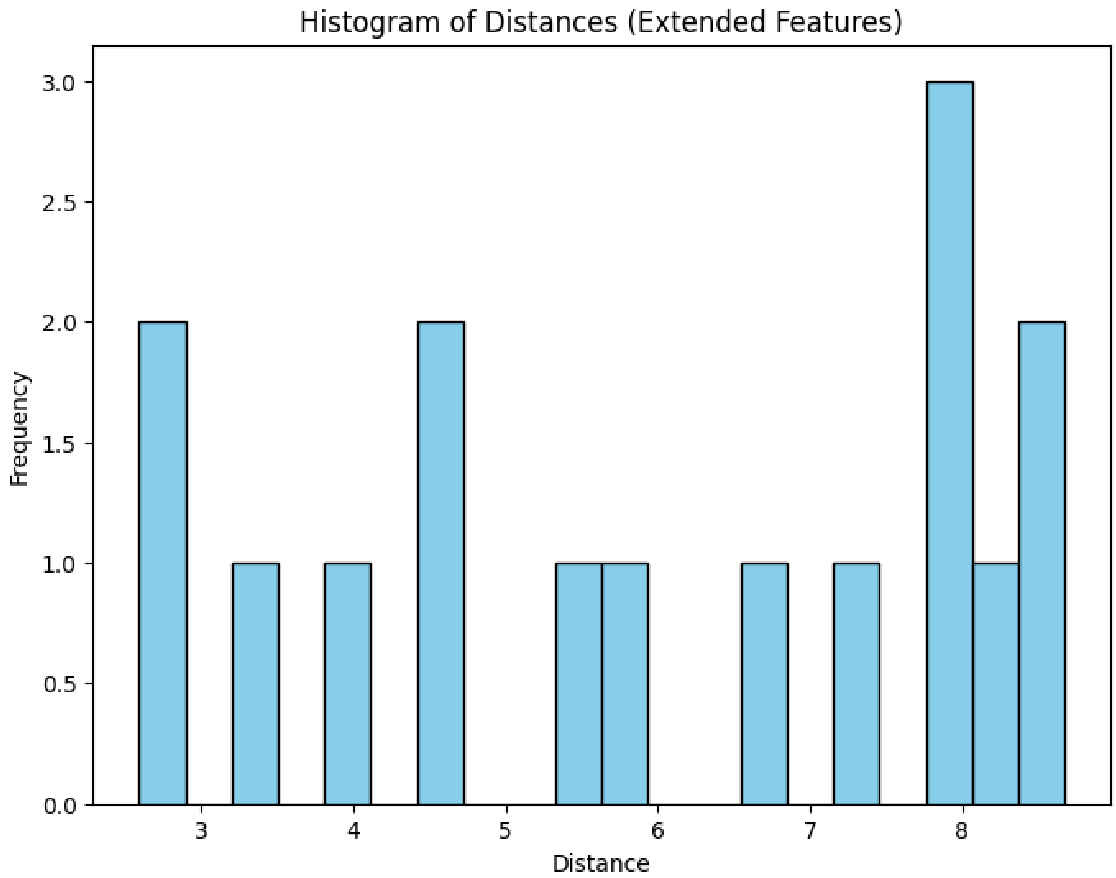

Figure 2: Distribution of Euclidean Distances with Extended Features

4.1.3 Classification Outcomes

Utilizing a threshold value of 6.22, the test signals were classified as follows:

Test Signal	Assigned Category	Distance
test_j	Peugeot_307_Drive	2.61
test_e	Unknown	6.74
test_n	BMW_1_Drive	2.59
test_m	Unknown	7.41

Table 1: Classification Results Using Combined Features (MFCC + Custom Spectral)

```
Selected distance threshold: 6.22

Classification based on combined features (MFCC + custom spectral):
- Test Signal test_j is categorized as Peugeot_307_Drive with a distance of 2.61
- Test Signal test_e is categorized as Unknown with a distance of 6.74
- Test Signal test_n is categorized as BMW_1_Drive with a distance of 2.59
- Test Signal test_m is categorized as Unknown with a distance of 7.41
```

4.2 Identified Recordings

Out of the four test signals, two were successfully matched to known car models. The inclusion of the "Unknown" category ensures that recordings not sufficiently similar to any reference are not erroneously assigned. Specifically:

- **Test Signal test_j**: Correctly identified as `Peugeot_307_Drive` with a distance of 2.61, well below the 6.22 threshold, indicating a strong similarity.
- **Test Signal test_e**: Classified as "Unknown" with a distance of 6.74, surpassing the threshold, suggesting it doesn't closely resemble any known model.
- **Test Signal test_n**: Accurately assigned to `BMW_1_Drive` with a distance of 2.59, significantly below the threshold.
- **Test Signal test_m**: Designated as "Unknown" with a distance of 7.41, exceeding the threshold.

4.3 Additional Visualizations

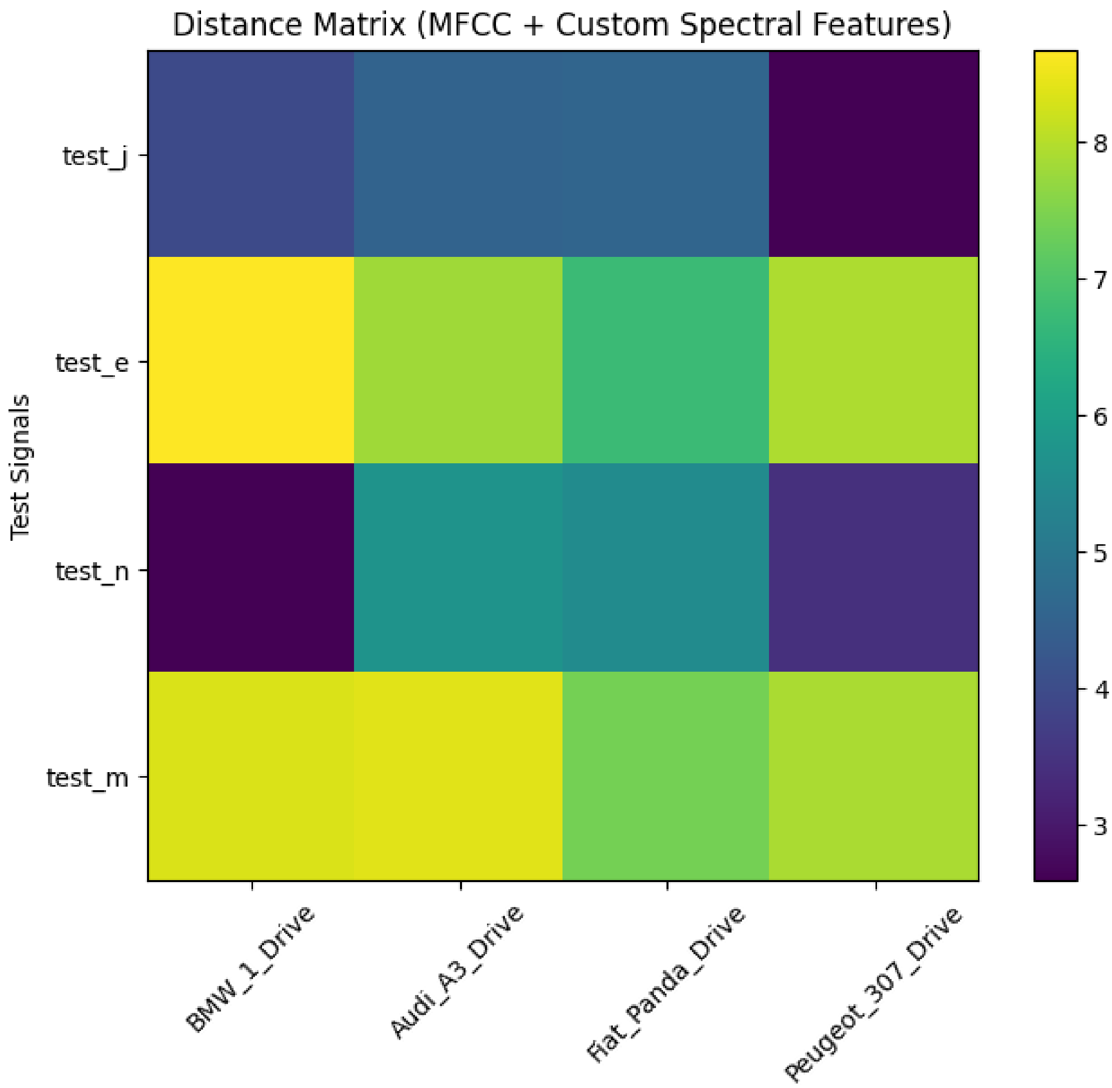

Figure 3: Color-Scaled Distance Matrix Comparing Test and Reference Signals

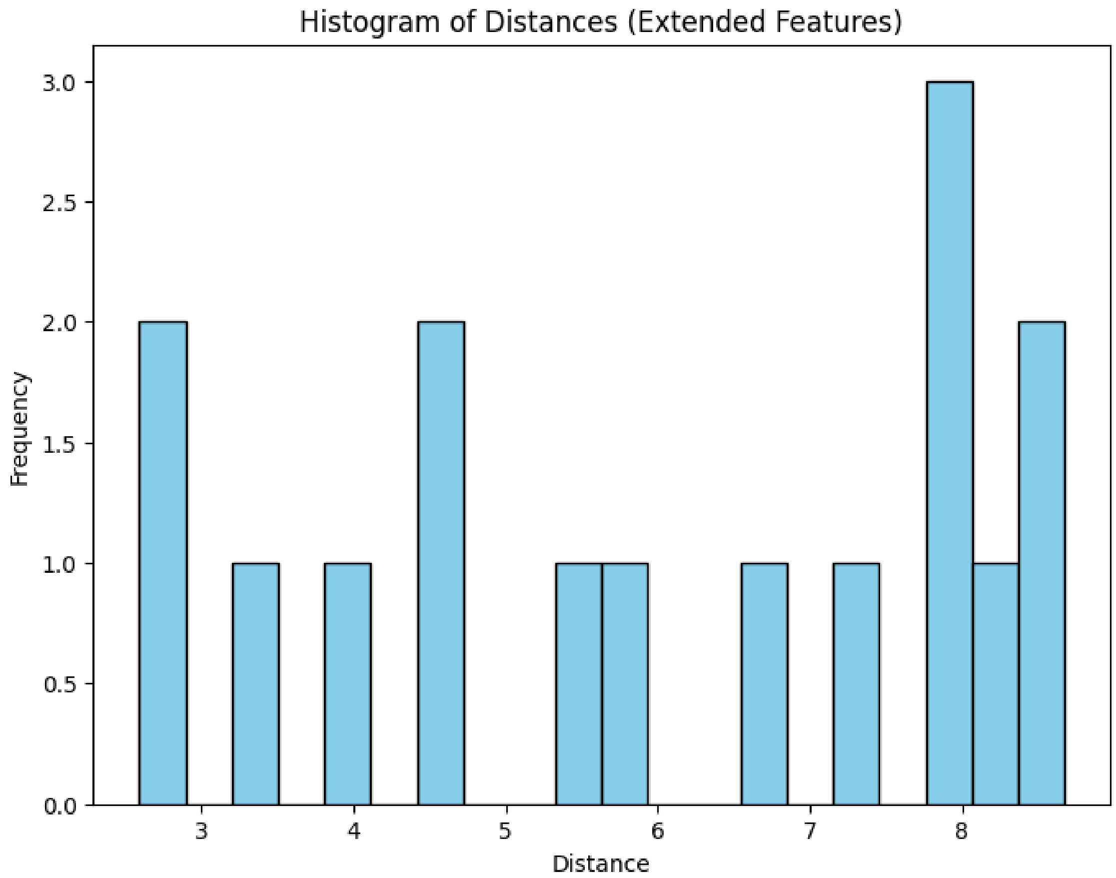

Figure 4: Histogram Illustrating the Euclidean Distance Distribution Between Test and Reference Signals

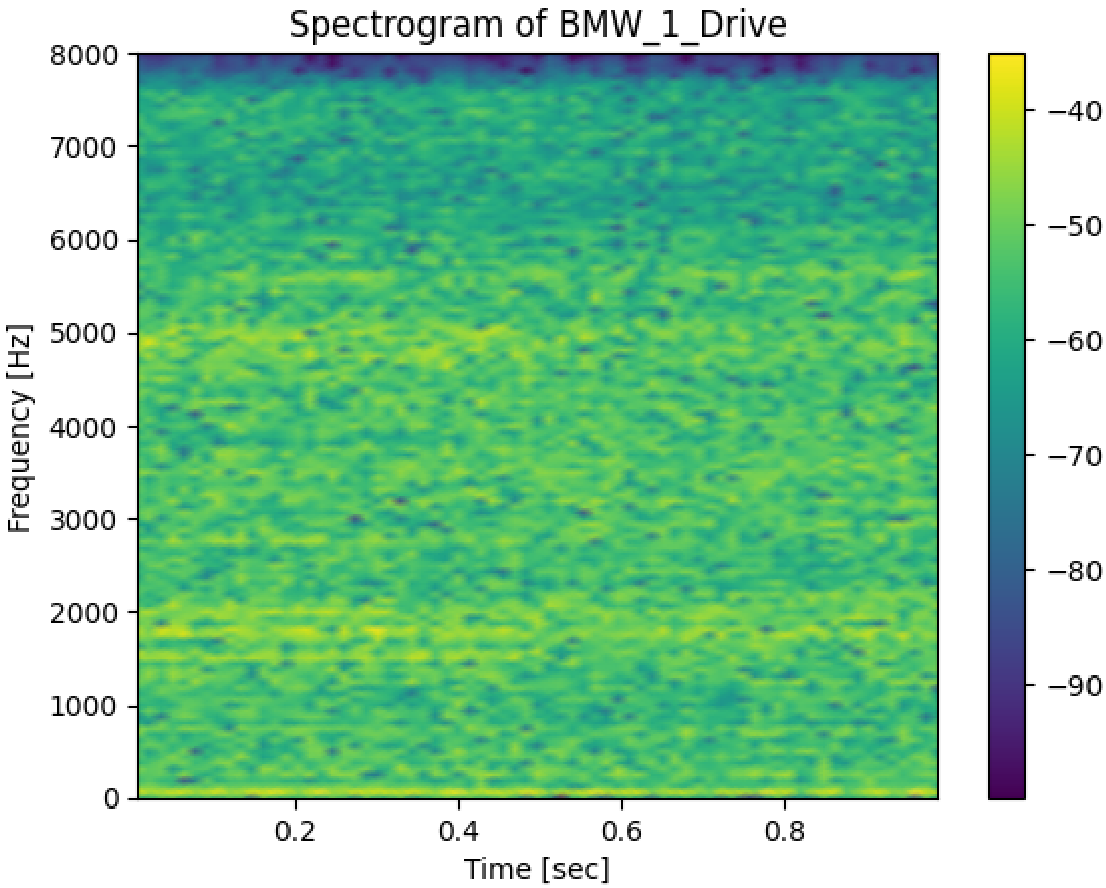

Figure 5: Spectrogram of one of the known signals

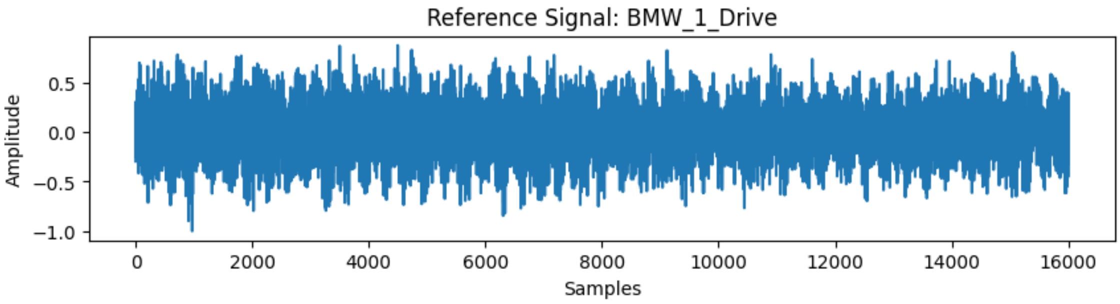

Figure 6: Spectrogram of the Reference Signal BMW_1_Drive

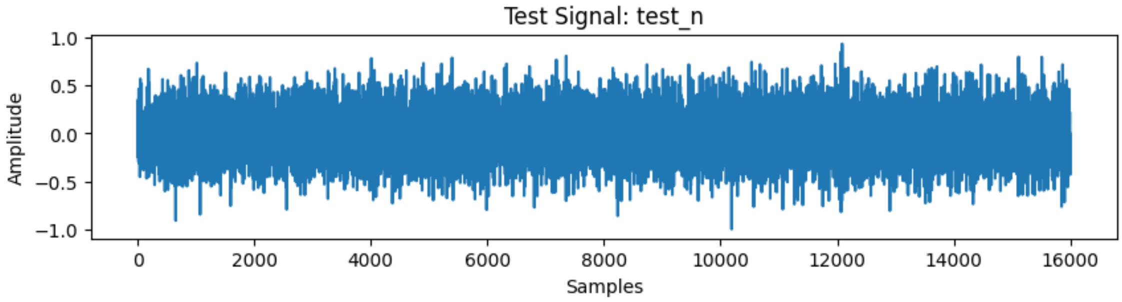

Figure 7: Spectrogram of the Test Signal test_j

Note: Figures 6 and 7 demonstrate example spectrograms generated by the system, highlighting the spectral analysis of both reference and test signals.

5 Conclusion

This project successfully classifying engine sound recordings from various car models. By integrating Mel-Frequency Cepstral Coefficients (MFCC) with additional spectral features, the system effectively matched test recordings to known models or categorized them as "Unknown."

6 References

https://colab.research.google.com/drive/1jftGwoJ2gEaTNjEnXBZRaY8XtqApD6Xw?usp=sharing