Introduction to audio and speech processing – project work

Tiia Jaskari, Sara Nurminen

Introduction

The aim of this project is to design and implement a binary classification model for sounds in audio clips, focusing specifically on classifying sounds associated with vehicles. The project involves a comprehensive workflow, starting from data collection to the final evaluation of the model's performance. This study addresses the challenge of distinguishing between two selected classes of vehicle sounds, leveraging audio signal processing and machine learning techniques.

The primary goal is to accurately classify audio signals recorded in real-world environments into one of two predefined classes, trams and cars. This involves not only the application of advanced algorithms but also critical decision-making in areas such as feature extraction, model selection, and data preprocessing. By the end of this project, we aim to assess the feasibility of sound-based classification using relatively simple audio features and models, while also identifying potential limitations and areas for improvement.

Data description

The data was collected on December 9th, 2024, in Hervanta. We selected two sound classes for the project: cars and trams. To gather the data, we recorded trams passing by while standing next to the tramway, followed by recording cars passing by near a road. In total, we obtained 23 samples of tram sounds and 20 samples of car sounds. However, two car sound samples were deemed unsuitable for this project due to background noise caused by people talking. Thus, 18 car sound samples were included in the analysis. The audio samples exhibit variation in loudness, primarily due to differences in the recording distance used during data collection. Our own data was used as test data and the train data was downloaded from freesound from other people in our course.

Feature extraction

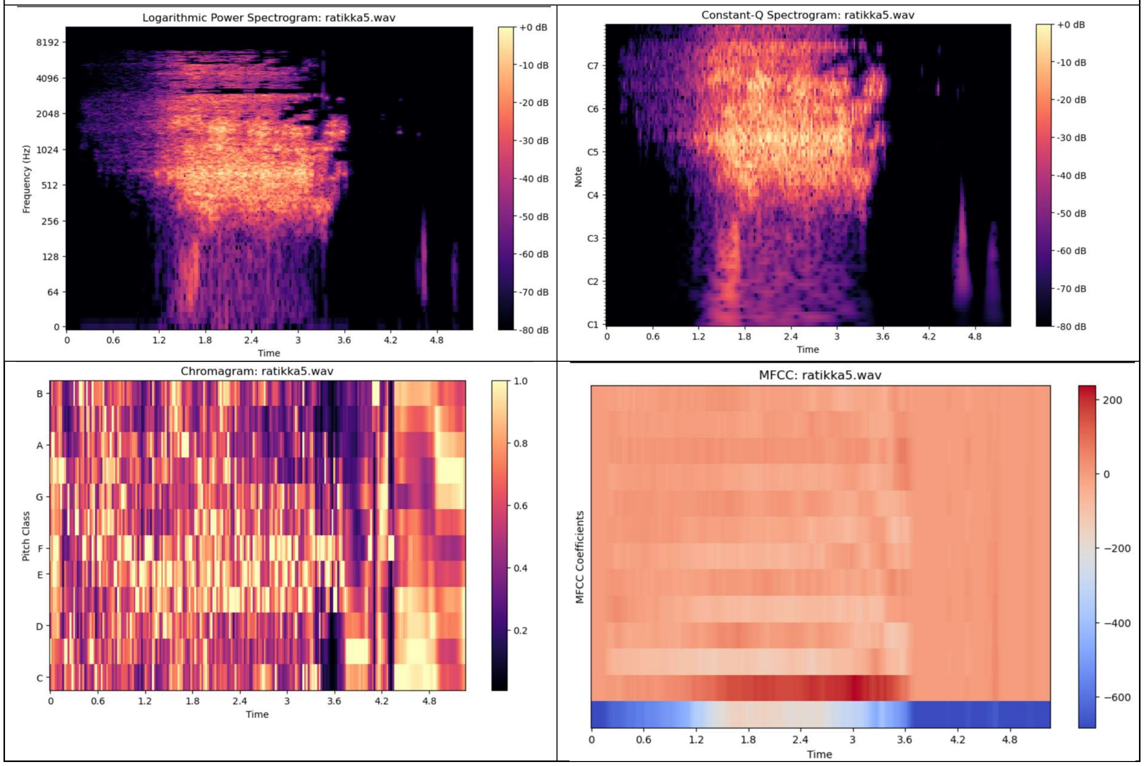
We chose the MFCC features for classifying vehicle sounds because they provide an efficient and compact way to analyze the low and mid-range frequencies that are typical of car and tram sounds. MFCC condenses the complex frequency spectrum into just 13 coefficients, making it easier to use with machine learning models and reducing computational load. We compared MFCC with the Constant-Q Spectrogram, Chromagram and Logarithmic Power Spectrogram. MFCC stood out particularly well identifying mechanical vehicle sounds, such as the hum of a car engine and the clatter of tram tracks.

The Constant-Q Spectrogram provides a detailed frequency resolution, especially for low frequencies. It might be good for music analysis but it did not perform well for mechanical vehicle sounds due to its sensitivity to noise and higher computational requirements.

Chromagram on the other hand features harmonic and tonal content by mapping sounds to pitch classes. However, vehicle sounds lack clear harmonics, making it less effective for distinguishing between car and tram sounds. The Logarithmic Power Spectrogram provides a detailed representation of the power distribution across frequencies. This might have been a good choice also but we decided to go with the MFCC.

Below, we have attached images of the feature extractions for one example each of tram ‘ratikka5.wav’ and car ‘auto4.wav’ sounds.

Feature extractions for ‘ratikka5.wav’



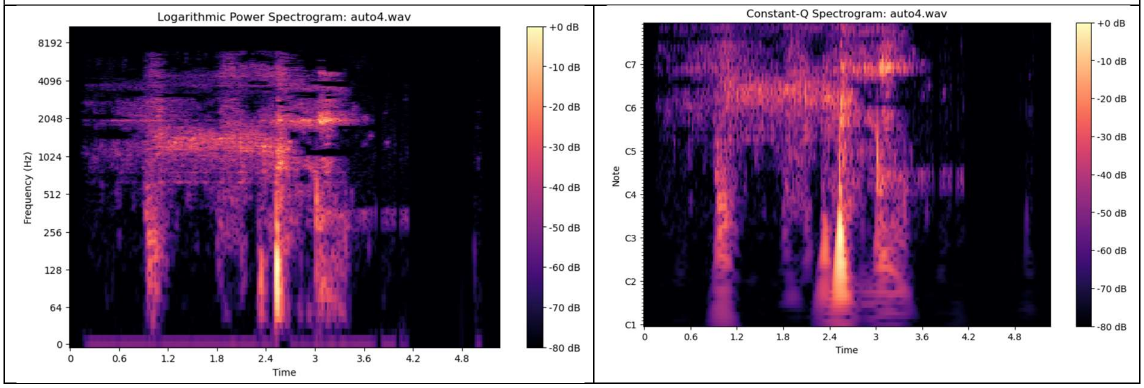
Logarithmic Power Spectrogram: ratikka5.wav

Constant-Q Spectrogram: ratikka5.wav

Chromagram: ratikka5.wav

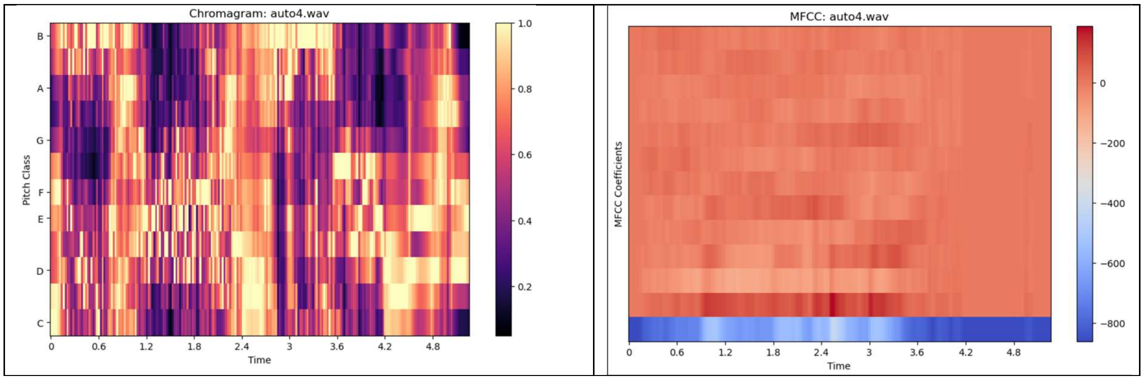
MFCC: ratikka5.wav

Feature extractions ‘auto4.wav’



Logarithmic Power Spectrogram: auto4.wav

Constant-Q Spectrogram: auto4.wav



Model selection, data split

We chose the Nearest Neighbors (K-Nearest Neighbors, KNN) model for our classification task due to its simplicity and because it was the only model we were familiar with. KNN is a non-parametric, instance-based learning algorithm that classifies data points based on the majority of their nearest neighbors. It is easy to implement and interpret, making it a suitable choice for our initial model.

In our final version, we used k = 1, which means the classification was based on the single closest neighbor. We observed that changing the value of k did not significantly impact the results, indicating that the model's performance was consistent across different k-values. The k-value determines how many nearest neighbors are considered during classification. Sometimes a small *k* can lead to overfitting, while a larger *k* can provide more generalized results.

Data was split into two groups: training data (car 33, tram 31) and testing data (18 car, 20 tram). The training data was as well split into two groups, actual training data consisting 70 % of the training data and the validation data consisting 30 % of the training data. The validation data was used to fine-tune the k-value and assess the model’s performance during development, helping to mitigate overfitting.

Results

Our model achieved an accuracy of 50%, which is not sufficient for it to be considered useful. While the model had an accuracy of 100% on the validation data, its performance dropped to 50% on the test data. This discrepancy suggests that the model may be overfitting to the training data, meaning it performs well on known data but struggles to generalize to unseen data. Essentially, the model's predictions are no better than random guessing.

The F1-scores for both the car and tram classes were low (0.46 for cars and 0.54 for tram), indicating that the model has difficulty distinguishing between these two classes. The model performs better at recognizing tram sounds than car sounds, as shown by the precision scores. Although the difference between the car and tram sounds is not substantial, there is a slight distinction that the model is able to capture to some extent.

Classification report with test data:

	Precision	Recall	F1-score	Support
Car	0.47	0.44	0.46	18
Tram	0.52	0.55	0.54	20
Accuracy			0.50	38
Macro avg	0.50	0.50	0.50	38
Weighted avg	0.50	0.50	0.50	38

Conclusion

This project was challenging for us, as we had no prior experience with machine learning. Our limited familiarity with different models made it difficult to select the optimal approach or improve our model effectively. Despite this, we gained valuable insights into machine learning fundamentals, audio feature extraction, and model development.

We chose the Nearest Neighbors (KNN) model due to its simplicity, but it proved to be less effective for our task. The similarities between car and tram sounds made it difficult for KNN to differentiate between the two classes. It could have been better to choose some other model to get better results.

Our feature selection relied primarily on MFCC (Mel-Frequency Cepstral Coefficients) due to their effectiveness in capturing low and mid-range frequencies. However, combining MFCC with additional features like Chromagram or Spectral Centroid could have offered a more detailed representation of the sounds and improved classification accuracy.

The data-handling process also posed challenges. Our training dataset was small (33 car, 31 tram), which limited the model's ability to generalize. Increasing the training data or applying data augmentation techniques (e.g., adding noise or altering pitch) could have improved performance. Additionally, the test data quality was poorer compared to the training data, particularly for car sounds. Using higher-quality recordings or publicly available datasets might have mitigated this issue.

Despite these obstacles, we learned a lot about the importance of model selection, data quality, and feature engineering. This experience provided a solid foundation for future projects, and we are eager to apply these lessons moving forward.