Binary Sound Classifier

Ella Rinnemaa

Nea Peltola

Introduction

This report covers the process of implementing a binary sound classifier. The aim of the program is to classify audio signals as either the sound of a tram or the sound of a bus.

Data Description

As is apparent from the first section, we chose to collect samples from two different classes, the sounds of trams and buses. We divided the workload so that one of us collected samples of trams and the other one of buses. The sounds of trams were recorded in Tampere using OnePlus Nord, whereas the sounds of buses were recorded in Tampere and in Helsinki using an iPhone 11. The sounds have been recorded during November and December of 2023.

Trams and buses have different stages of motion, which can be categorized as acceleration, deceleration and stopping. Additionally, buses can idle whereas trams can be stationary. To provide an encompassing data collection, we aimed to record and gather samples from all the different stages. Most of the sounds from buses have been recorded at a bus stop where many of the motion stages occur and the recording distance is small. The tram sounds have been recorded next to the tracks where the tram passes by with average speed, as well as at the tram stop where the tram slows down and speeds up.

Overall, we recorded 13 samples from buses and 15 samples from trams with varying number of sounds from different stages of motion.

Data used for training and validation is data provided by other students on the course.

Feature Extraction

For feature analysis, we used five features: Energy, Root Mean Square (RMS), Spectrogram, Mel Frequency Cepstral Coefficients (MFCC) and Constant-Q Spectrogram (CQT). We used different features to distinguish different characteristics of the data classes, tram and bus sounds.

Energy and RMS represent the intensity of a signal, capturing overall loudness and energy distribution. With our test samples, the bus and tram histograms resembles each other with both energy and RMS values (Attachments, Picture 1 and Picture 2) which is why we did not use them in the model.

Spectrogram provides a time-frequency representation of the audio signal, showing how frequency of a signal changes over time. Our bus and tram samples do not have very distinct frequency distributions (Attachments, Picture 3) which is why it is not a suitable feature to use in the classifier model.

Constant-Q spectrogram shows a logarithmically spaced frequency representation, mirroring human auditory system where the resolution for lower frequencies is better. CQT feature is valuable for classification where the pitch variations are important, for example with music audio. With our bus and tram audio samples, the CQT is not different enough between the samples (Attachments, Picture 4) which is why we decided to not use it in the model.

MFCCs represent spectral characteristics of a signal, capturing features like pitch and timbre of the audio signal. MFCCs represent signals in a manner equal to human auditory perception, which is why it is an excellent feature to use in a classifier. MFCCs for our samples is represented in Attachments, Picture 5. We chose MFCCs as the feature used for the classifier because of its ability to capture differences in vehicle audio samples. The feature also works well with the classifier model we chose.

Model Selection, Data Splitting

We chose to implement the classifier using a Support Vector Machine (SVM). SVM is a supervised machine learning algorithm, that is particularly well-suited for small- or medium-sized datasets. The primary goal of an SVM in classification is to find the best separating decision boundary (hyperplane) that divides different classes in the feature space. This hyperplane is chosen to maximize the margin, which is the distance between the hyperplane and the nearest data points from each class, known as support vectors. One of the key features of SVM is that it can handle nonlinear relationships. This is achieved with kernel functions (like polynomial kernels) that transform the input data into a higher-dimensional space where a linear hyperplane can be used to separate the data.

SVM was chosen because it corresponded with our needs: a simple algorithm that is suitable for small datasets. Furthermore, we weren’t too familiar with SVM, so studying its theory provided a learning experience.

In our code we used scikit-learn's support vector machine –library's class SVC (Support Vector Classification) and provided it with a polynomial kernel function. Using a polynomial kernel function provided better results in comparison to a linear kernel function. For evaluating the model, we used scikit-learn's metrics -library.

In data splitting we utilized the disjoint data splitting strategy, which was introduced in the project guidelines. For test data we used our own samples in folders 'bus_samples' and 'tram_samples'. For training the model we used data which other students on the course provided on FreeSound. These sets are in folders 'training_bus' and 'training_tram' and were provided by users aleksin, samulihynninen, araatikainen, akahukas, lndip, aaroan, marjiaakter, mslv, masa_ite, julianschur and hnminh. We utilized other students' samples also for validation data: user's thespicychip's packs '39926__thespicychip__tampere_bus_audio_data' and '39927__thespicychip__tampere_tram_audio_data'.

Results

Table 1: Results using SVC with polynomial kernel function.

	Accuracy score	**Precision score**	**Recall score**
Training	92%	87%	99%
Validation	70%	85%	59%
Testing	64%	71%	38%

The results vary across training, validation, and testing phases. Training stage has high accuracy meaning that the model performs well on training data. Training precision is good, suggesting that the model's class predictions are correct most of the time. Recall is excellent so the model successfully identifies almost all relevant instances. Validation stage shows a significant drop in the accuracy while precision remains high, meaning the model's predictions are consistent. Recall score is also significantly lower, indicating that the model misses a considerable number of relevant instances. Testing accuracy is low, which when compared to the results in the training stage, indicates overfitting issues in the model. Precision score is okay, though it still shows a decrease in predictive correctness. Recall score is poor showing that the model fails to identify majority of relevant instances in the testing stage.

Conclusion

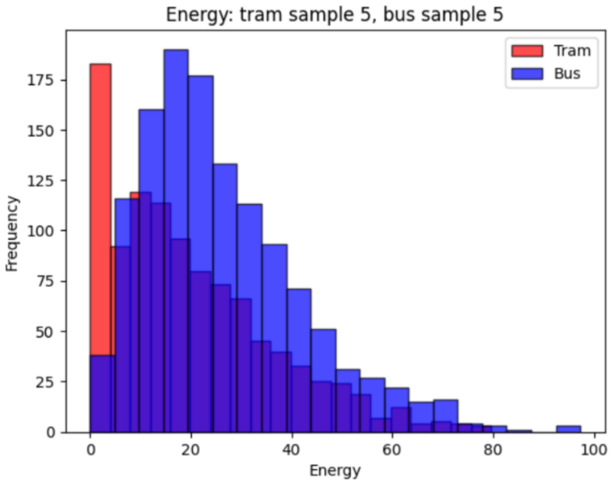
The major difficulties affecting the results are related to the audio samples. The samples were recorded with phones which have poor microphone quality compared to a real microphone. Also, the recording was done at a bus or tram stops or next to a street where the microphone easily captured background noise or even noise from other vehicles and people talking. The audio samples also experienced quite a lot of compression as the original file format was converted into wav, uploaded to FreeSound, downloaded to computer, and then loaded into the python program with Librosa.

Another issue is the overfitting in the model. The model performs well on the training data but struggles to generalize to unseen data, as can be noticed from the performance drops in validation and testing. The significant drop in accuracy and recall from training to testing suggests that the model might not be generalizing well to new, unseen data.

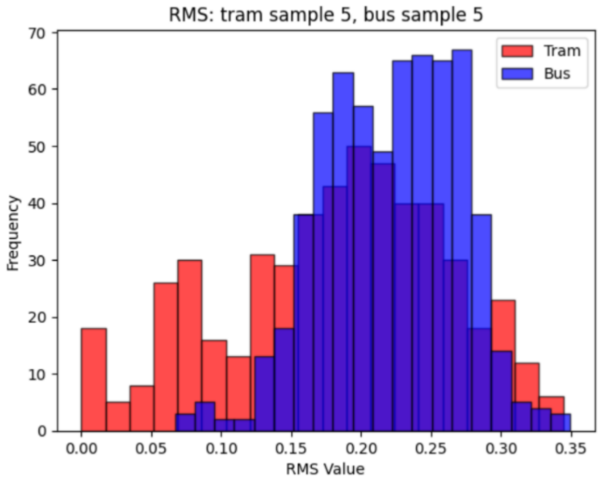
Other issue affecting the results is the small amount of test data. We were only able to collect quite few samples of busses and trams which can be the cause of less accurate classifier model.

Overall, our classifier is still a decent model to distinguish tram and bus sounds from each other and estimate the correct class for an audio sample. The workload in this project was equally divided. We both collected audio samples and then divided the coding work that Nea focused on the features and Ella focused on the classifier. The report was written by both and we kept close contact throughout the process.

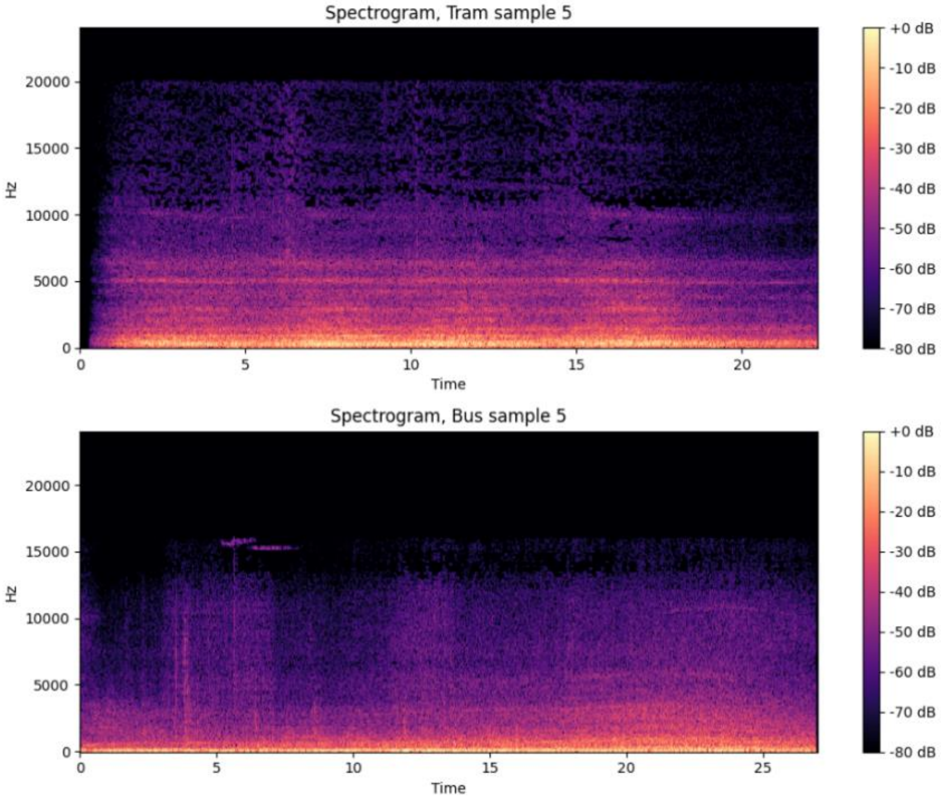
Attachments



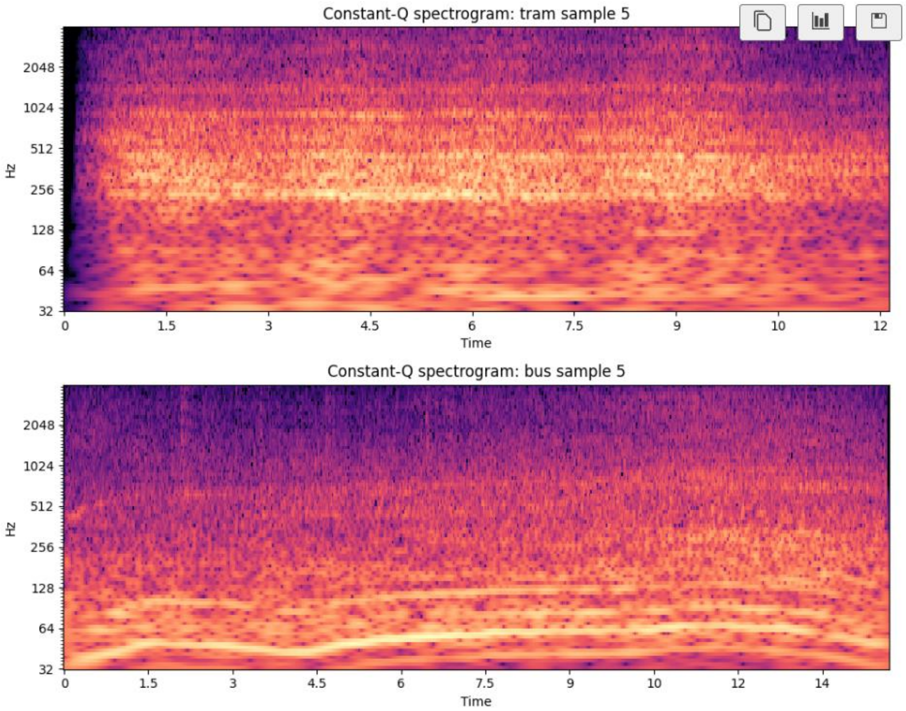
Picture 1 Energy for tram and bus audio samples



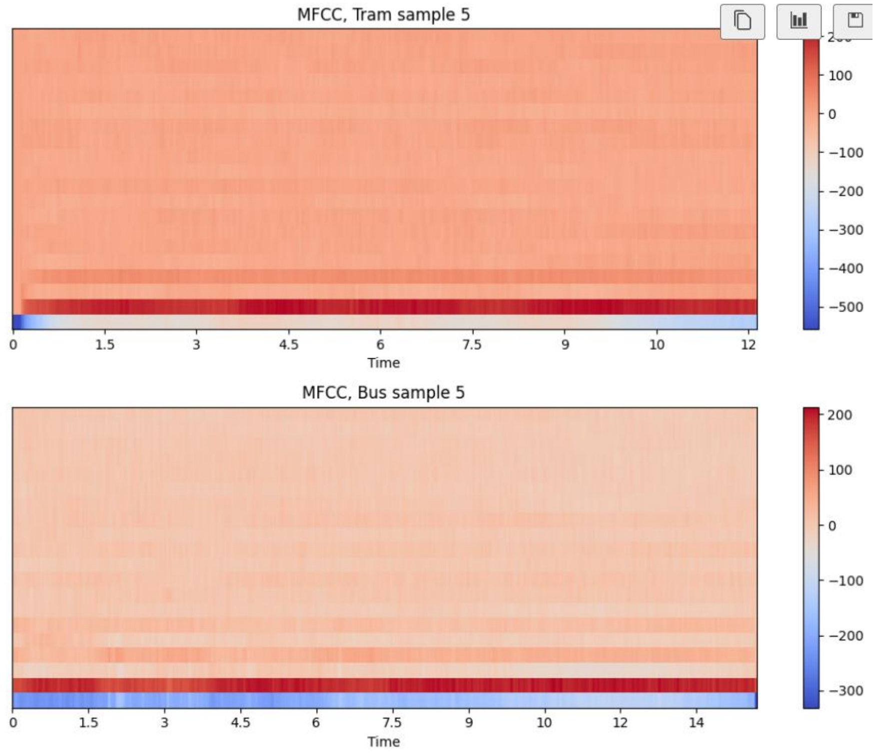
Picture 2 Root Mean Square for tram and bus audio samples



Picture 3 Spectrograms for tram and bus samples



Picture 4 Constant-Q Spectrograms for tram and bus samples



Picture 5 Mel Frequency Cepstral Coefficients for tram and bus samples