**Introduction**

People with impairment residing in Tampere may have it difficult to know when the tram is comming in a tram stop. Our goal is to build a Binary classification model to detect if a given sample is an audio of a car, or a tram. This could also help in engineering systems that aim to help people with impairment in this situtations.

**Data Description**

The data used was 30 samples of 5 seconds max. for each class, we have 2 classes: trams and cars, so a total of 60 samples were used. The samples are a mix of audios collected by us with our phones, and audios from freesound.org with Creative Commons zero. Both, the samples collected by us and the downloaded ones, were inspected by ear to have a low-noise sample set.

**Feature Extraction**

We trained the model with 7 features from the collected audios. The mean of each feature was taken to be able to train the SVM model.

1) Chroma features: represents the energy of each pitch over time, which highlights the melodic and tonal content of the audios. Each object usually have its own characteristic pitch; as car audios exhibit noise of more parts, this usually leads to a sparse chroma pattern, while tram audios have a more consistent tonal content. Also, as the trams of Tampere are always the same model, the pitch pattern is the same always.

2) Tonnetz (tonal centroid): similar to Chroma, Tonnetz represents the energy of each of the main harmonic intervals over time (perfect fifth, minor third, and mayor third).

3) MFCC: the Mel-frequency cepstrum coefficients highlights the time-frequency energy distribution of the audio samples. While car audios tends to have a broader energy distribution with high activity on 0.50-5.00 kHz and some irregular noise because of the honks or engines; tram audios tend to have the energy concentrated in the range of 0.05-0.50 kHz and also a smoother and more periodic pattern.

4) Spectral centroid: frequency domain feature that indicates where is the center of mass of an spectrum, it is calculates by taking hte weighted mean of the frequencies of a signal. It predicts the “brightness” of a sound, and is a good parameter to measure musical “timbre”. Trams will have a very different timbre compared to cars, as a violin have a different timbre compared to a guitar.

5) Spectral contrast: is the difference in decibel between peaks and valleys for each frequency subband in the spectrum

6) RMS: time-domain feature

7) ZCR: time-domain feature

**Model Selection**

We decided to use a Support Vector Machine (SVM) with a linear kernel. This is because SVM is, in principle, designed for binary classification of small datasets with a clear margin of separation, although it may be worse when dealing with noisy data. The linear kernel was chosen over more complex models because of the expected clear difference between the chosen classes, and to try to reduce overfitting. Hyperparameters like C or Gamma were not tunned for this model. Regarding the data set, it was first normalized using a Standard Scaler, this prevents any single feature to dominate the learning process, It works by ensuring that each feature have a mean of zero and a standard deviation of one. The data was then split into training set (60%), validation set (20%) and test set (20%).

Additionally, a noise test was performed, in which we applied noise of 5% and 10% to the Test set, and then try the model with it, an then calculating accuracy, precision, and recall.

**Results**

A table presenting the accuracy, precision and recall results of the data is presented below

**Table 1:** validation, test, and noisy test results of our SVM model.

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Set** | **Accuracy** | **Precision** | **Recall** |
| Validation | 1.00 | 1.00 | 1.00 |
| Test | 1.00 | 1.00 | 1.00 |
| Noisy Test (5%) | 0.85 | 0.89 | 0.80 |
| Noisy Test (10%) | 0.67 | 1.00 | 0.33 |

Our model can correctly classify if a given audio is a Tampere Tram or a car passing through. The only inconvinience is that if a it can only works with 100% accuracy if it is a Tampere tram or car audio with minimum external noise. With 5% noise, our model still works well, but with 10% noise, the model starts to be too inaccurate. As this is a problem of binary classification, we assume that a false positive is as costly as a false negative, the accuracy test is the most relevant, as it assigns equal weight to false positive and false negatives. We think that this is maybe caused by overfitting in the model, as we used a linear SVM with 7 features. But this is not necessarily bad, as it makes the model perform great with similar data (trams in Tampere).

**Conclusions**

Perfect scores on the normal data test suggest either possible overfitting, a trivial soluton for the separation of the data. The low scores on the noise test further suggest possible overfitting, but it could also be caused by the noise masking our selected features.

Some solutions to this problem might be applying noise during training to account for the possible noise, that is very common in most real life situations.It might be that training the model with different features could solve the problem, but we think that the problem is not about the features, but rather a combinaton of overfittiing and that the separation of the data is trivial.

Additionally, tunning the C and gamma hyperparameters by cross validation or grid search might help with making a better model for these situations.

Regarding the features used, we think that the number of features is too high realtive to the dataset size, and this increments the risk of overfitting while training the model. This could be overcomed by excluding the less relevant features, using techniques like Principal Component Analysis or Recursive feature elimination.