AUDIO AND SPEECH PROCESSING PROJECT REPORT

Joonas Kelavuori & Tommi Salonen

INTRODUCTION

The purpose of this project was to figure out how to make a good classifier for differentiating tram audio from car audio. We felt trams would have a different enough frequency distribution to cars due to trams squealing when they pass by. We believed there to be clear frequency magnitude peaks in higher frequencies when looking at trams and indeed there was.

Both group members recorded audio samples, the python code was made with pair-coding method, Tommi wrote the code and Joonas gave improvement ideas. The documentation was written together. All in all work was divided as evenly as possible.

DATA DESCRIPTION

All the audio samples collected by us were either from cars driving on the road Hervannan valtaväylä in front of Tietotalo or trams passing by tram stop Opiskelija on road Insinöörinkatu. 38 car samples were collected while 37 tram samples were collected all in wav format. 83 tram samples and 104 car samples were downloaded from Freesound. A total of 262 audio samples were used in making this project.

By the nature of the data, it was self-explanatory to have two classes in the handling of the data. One for trams and one for cars.

FEATURE EXTRACTION

From the very beginning we had a good feeling about using some sort of frequency domain feature for this task since we knew that trams would have some sort of high frequency peaks much more pronounced than cars would have. This can be observed from figure 1.

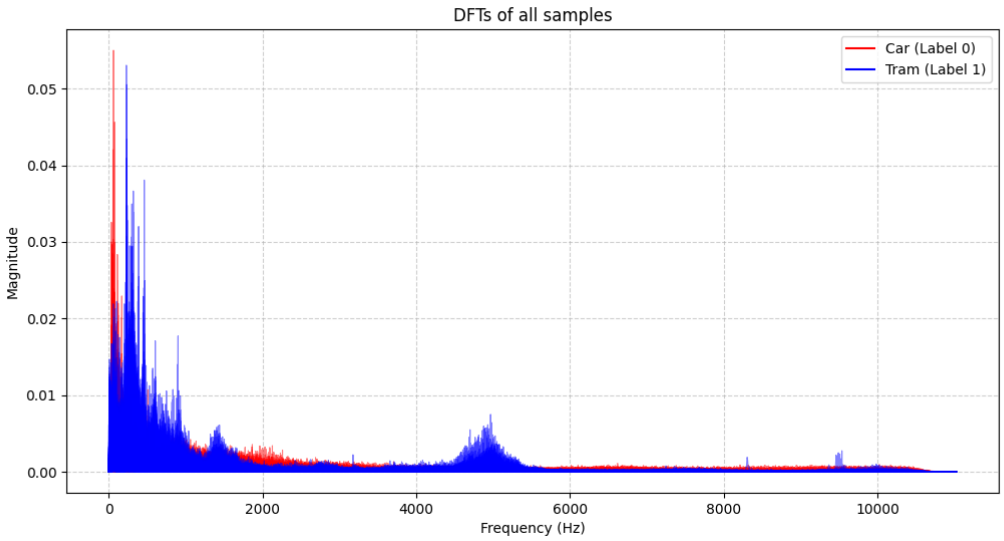

Figure 1. Frequency components of all the samples

We firstly did a DFT magnitude nearest neighbor model from an intuitive feel from which we got pretty good accuracy, precision and recall values in the neighborhood of 0.85 – 1. After that we took a second look at the instructions form the Moodle page (we had just been looking at the instruction pdf) and realized we needed 3 more features.

After this we downloaded more data for our model and we used MFCC, spectral centroid, rms energy, and our DFT magnitudes as our features. This led to bad accuracy precision, and recall values in the neighborhood of 0.45 – 0.7 with various k values in our nearest neighbor model. Using only our DFT magnitudes after getting more data led to very bad results. The DFT magnitudes array had such high dimensions that the classifier was having issues with it.

Then we decided to consult ChatGPT for what could be the best features based on our good result with our DFT magnitude feature. As an answer we got RMS energy, spectral spread, and spectral centroid with our DFT magnitudes. Spectral centroid tells where the most of the energy of signal concentrates in frequency domain. Spectral spread tells how wide the frequency spectrum of a signal is. RMS energy tells the average energy of the signal. Using them led to good accuracy, precision and recall values. Accuracy was 0.81, precision 1, and recall 0.64.

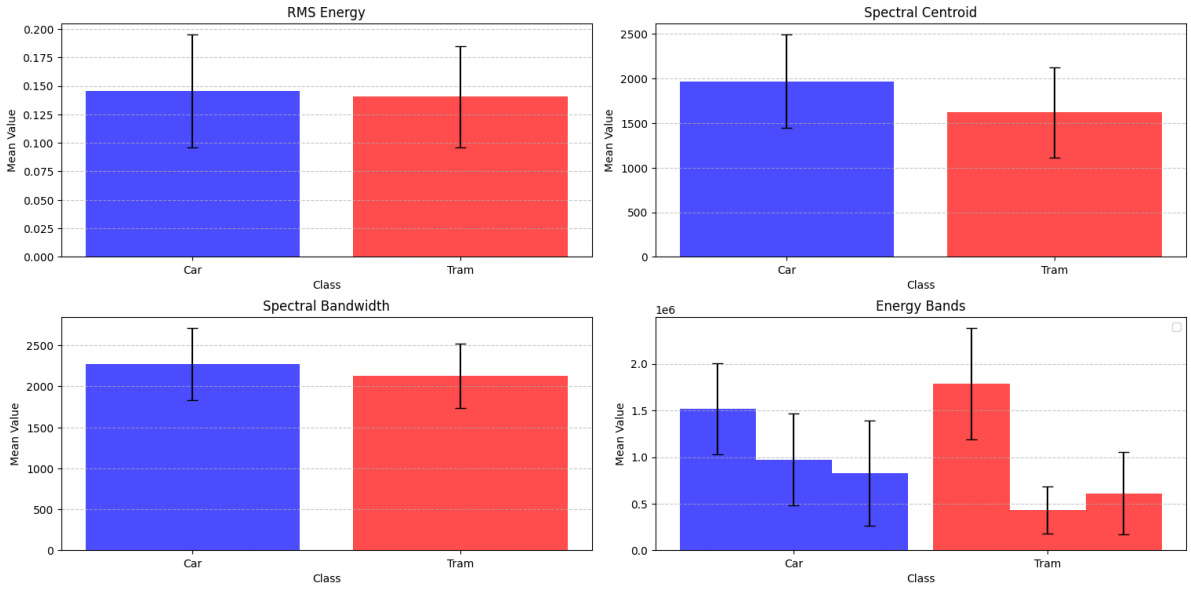

Figure 2. Comparing features

We decided to use the DFT magnitudes as our audio feature in the end. We split the frequencies into low-band- (0-1000 Hz), mid-band- (1000-3000 Hz), and high-band (3000-8000 Hz) average energies. With this we got good results for accuracy, precision and recall. RMS energies, spectral centroid, and spectral bandwidth mean value graphs for tram and car were much worse looking (similar) when plotted compared to DFT magnitudes as can be seen from figure 2.

MODEL SELECTION, DATA SPLIT

We decided to use a KNearestNeighbour model since we had prior knowledge of the model, and we felt like it was the most straight forward of the selection.

Samples were randomized to disjoint users, and the data was split so that 80% of the samples was used to train the model, 10% was used for validation and 10% was used for testing.

RESULTS

After training the model and testing it with different K-values and the validation data, we decided to use 11 as the value for K.

Table 1. Results

	Accuracy	Precision	Recall
Validation data	96.2%	90.9%	100%
Test data	88.8%	85.7%	92.3%

As can be seen from table 1, our classifier got good results. The classifier was able to determine which sounds were from a car and which from a tram, at a reliable rate.

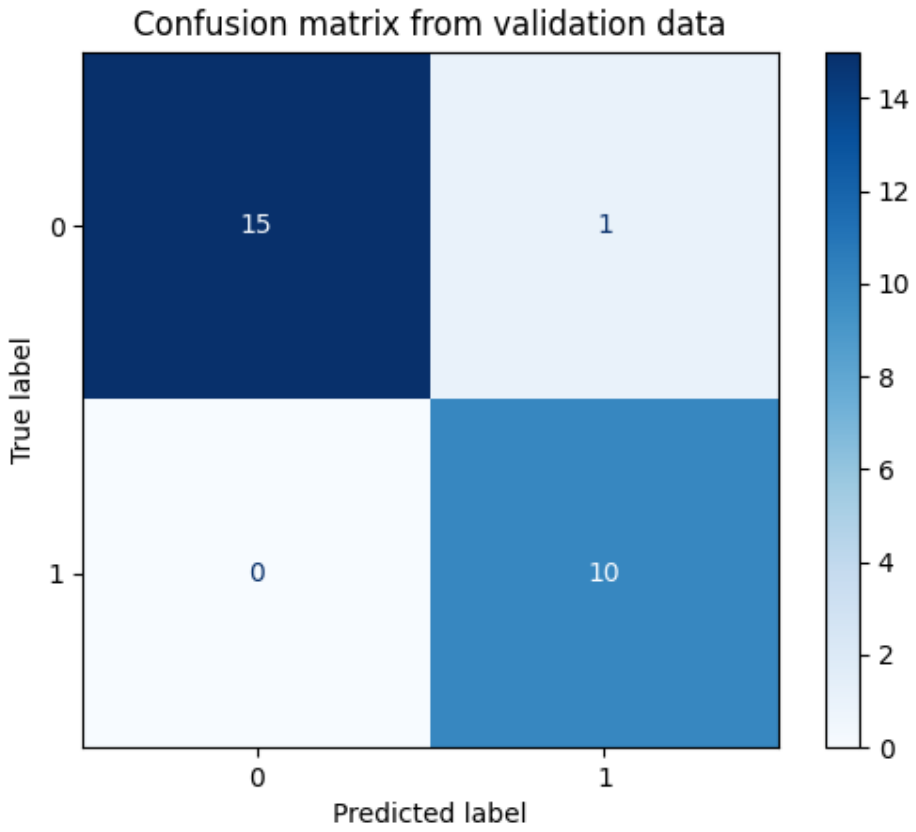

Figure 3. Confusion matrix from validation data

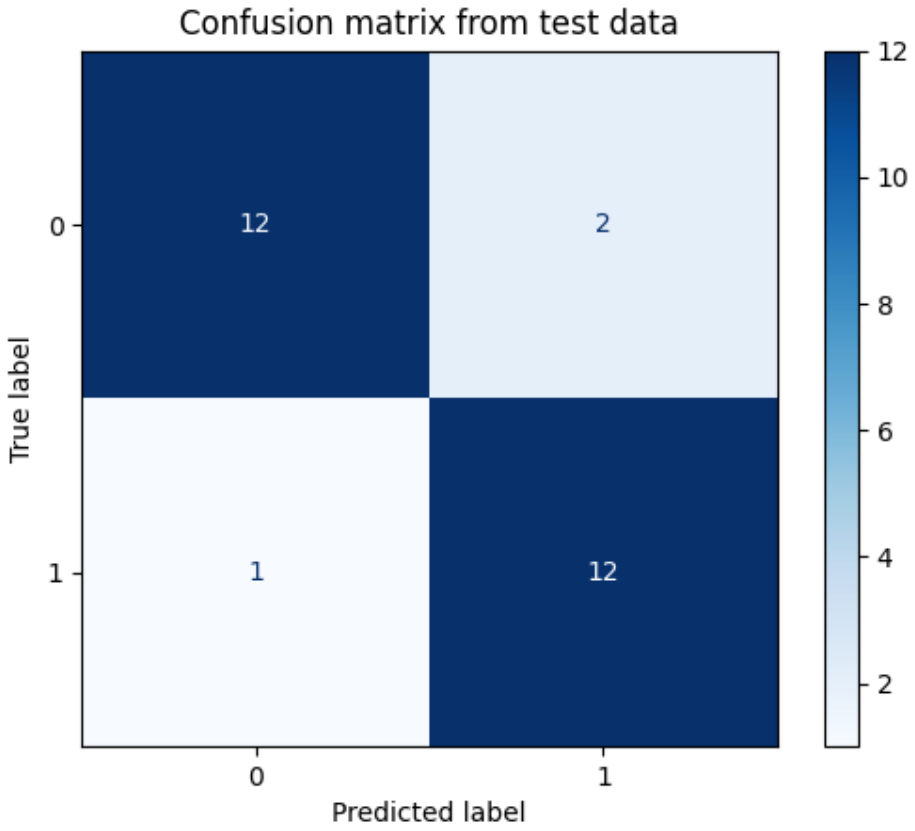

Figure 4. Confusion matrix for test data

From figures 3 and 4 we can see what the samples were predicted to be and what they were and that the validation data had more cars than trams in it, but in the test data the split between them was as even as possible. This might affect the results a bit.

CONCLUSIONS

As we first did our DFT magnitude nearest neighbour, it worked fine, until we downloaded more data. After we split the DFT magnitudes into three frequency bands we started getting good results. As both classes had higher magnitudes in the lower frequencies the high frequency peaks in tram samples, which were the key to differentiate the classes, were not getting as much weight as needed before we did the split.

We believe that the results could be better with a larger dataset, but we also believe that this dataset was enough for this project.

In the end cars and trams had different enough frequency distributions. It was in the end easy enough to make the classifier with very little experience in signal processing.