ELECTRICAL CHALLENGE 1 - REPORT

Audio Explorers challenge 2023

Authors:
Kaj Mørk
Benjamin Musak Hansen
Axel Villads Burford Toft

Aalborg, Denmark
April 2023

GitHub Repository: `https://github.com/BenMusak/Audio_explores_2023_AVS`

1 Introduction

The aim of this challenge is to develop a sound scene audio classification system for hearing aids, utilizing a provided training and test set. After exploring the dataset, we evaluated three different algorithms for the classification task: K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Convolutional Neural Network (CNN). The aim is to find the optimal trade-off between size, performance, and computational load.

2 Data Exploration

The training dataset provided consists of 52890 Mel-spectrograms, each corresponding to a two-second sound clip. These spectrograms represent five different classes; music, human voice, engine sounds, alarm, and others. Examples of each class shown in Figure 1. The dataset provides a comprehensive and diverse set of audio samples for training and testing, facilitating the development of effective sound recognition algorithms.

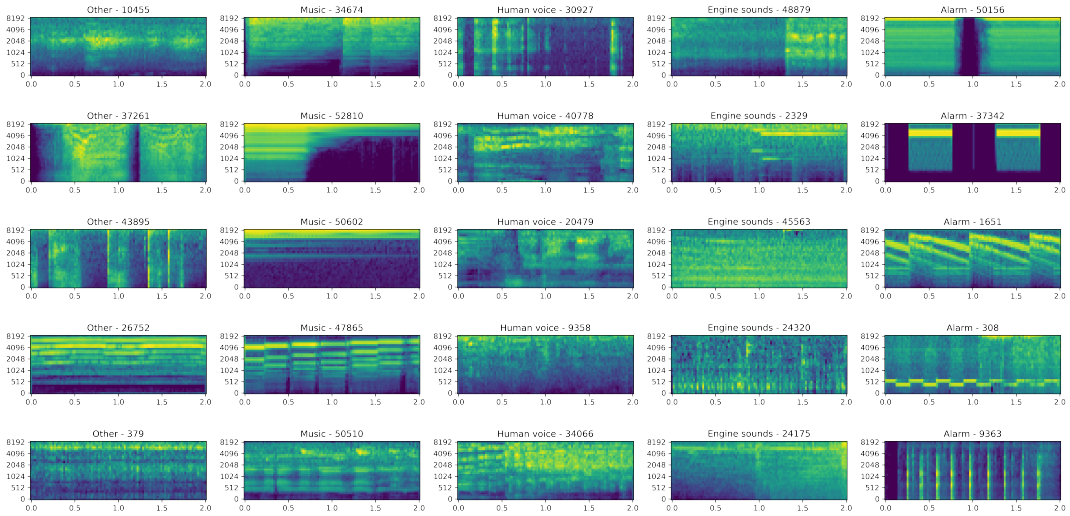

Figure 1: Samples from the 5 different classes.

The training dataset exhibits an uneven distribution of class labels, with the Music class comprising the majority at 51.69% (27340 samples) of the dataset, while the Alarm class has the lowest representation at only 3.37% (1785 samples). See Figure 2 for the distribution of the classes. The class label distribution in a dataset should represent the class' distribution in reality, as the distribution can significantly impact the performance of a machine learning model, by leading to biased predictions.

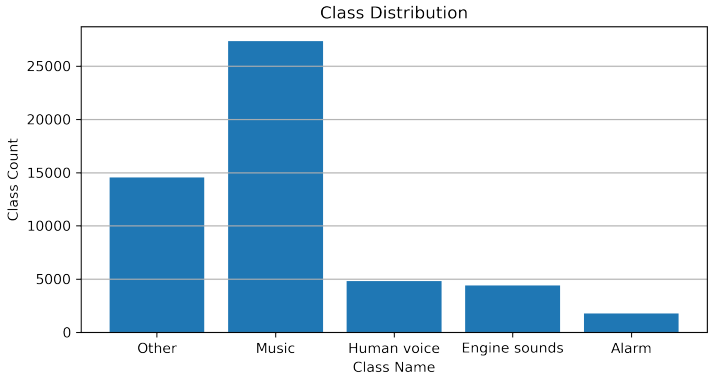

Figure 2: The distribution of the classes.

As well, an unlabeled testing dataset is provided, consisting of 5347 spectrograms, with the same properties as the training dataset.

3 Model Delimitation

We decided to do initial tests on the three chosen algorithms to decide which one should be part of the final solution. All tests was performed on a 12th Gen Intel(R) Core(TM) i5-12600K CPU and a RTX 3070 Ti GPU, with the dataset split as followed: 80% for training, 10% for testing, and 10% for validation.

3.1 KNN results:

The KNN was trained using Mel-Frequency Cepstrum Coefficients (MFCC), which were derived from the Mel Spectrogram, as well as the delta and delta-delta of the MFCCs. This was done because of the MFCCs strengths in recognizing the structure of a musical signal, modeling the subjective pitch, as well as modeling the frequency content of an audio signal [1]. To reduce the amount of features for a faster and smaller-sized model, the coefficients were summed in the time-dimension. This resulted in a model with the size of 9.85 megabytes with an inference time of ≈1.6 milliseconds. The accuracy of the training set is 54.2%, and the accuracy of the validation set is 53.8%.

3.2 SVM results:

The Support Vector Machine model was trained on a scaled version of the Mel-spectrogram, resulting in a model size of 593 megabytes.The accuracy achieved with this model is 82.6% for the training set and 78.5% for the validation set. However, due to size limitations, the model was also trained on just 1% of the dataset, resulting in a size of 9.1 megabytes. The accuracy achieved with the reduced size dataset was 74.9% on the training set, and 57.7% on the validation set.

3.3 CNN results:

The training was performed on a 4-layered CNN with a total size of 5.92 MB and 491,909 parameters. The model had an accuracy of 87% and and average inference time of ≈2 milliseconds. Furthermore, a PR-curve and confusion matrix can be seen in Figure 3.

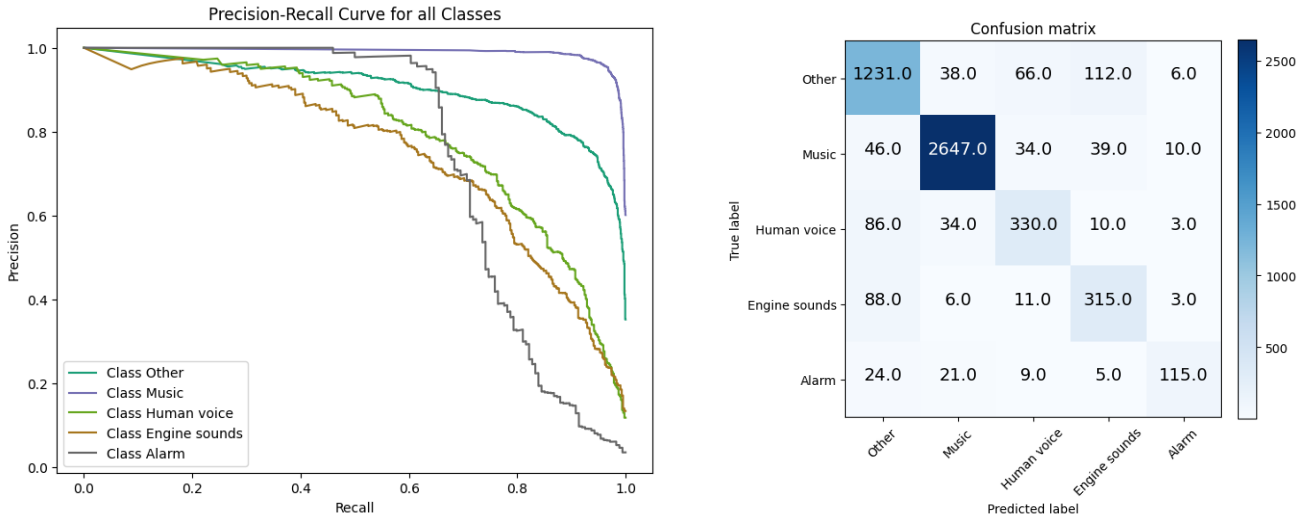

Figure 3: PR-curve and confusion matrix of the baseline CNN

3.4 Choosing a model

Looking at the results of the two models and the KNN, the CNN showed the highest potential with not only the smallest sized model but also the highest accuracy. While the SVM model showed some promise as well, with an accuracy of 82.6% and 78.5%, the size of the model had to be decreased to only 1% of the dataset to reduce the model size, which resulted in a lower accuracy than the CNN model. The KNN only showed some promise in its size and inference time, but quickly falls behind on accuracy. Therefore, the CNN model was chosen to be improved upon.

4 Optimizing the CNN

In the final iteration we took into consideration that hearing aids are limited in terms of memory and computational power. Doing so, we assumed that the given number of 500000 parameters is a max limit and that storing two second audio clips is not feasible on hearing aids. Furthermore, using smaller audio clips would also allow the hearing aid to respond faster. Taking this into consideration, we then implemented a three layer CNN with an expected input of 36x12, a total of 56,837 parameters, with a total model size of 0.70 MB (see Figure 4). The input of 36x12 is specifically chosen as this results in a 0.25 second audio clip. Since we were provided with two second spectrograms, we could then split them up into 8 36x12 spectrograms, and extend the labels to compensate for the added number of spectrograms.

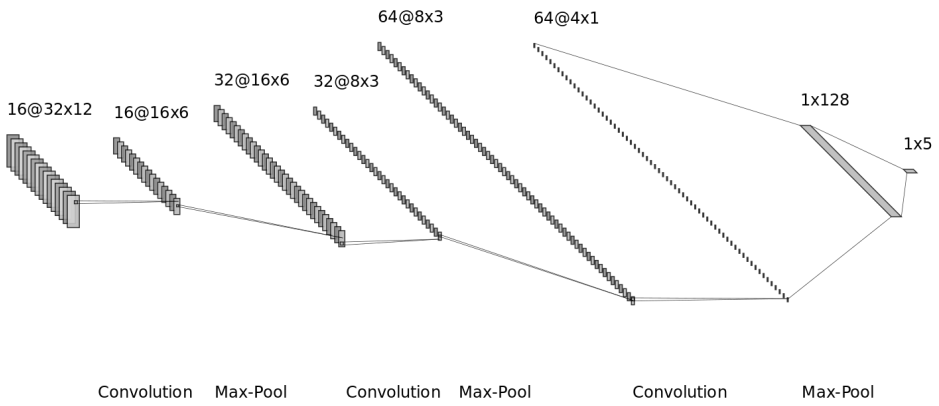

Figure 4: Used CNN architecure

Results after training shows a slight decrease in accuracy from 87% to 85% on the validation dataset, but also a decrease in average inference time, from approximately 2 milliseconds to 1 millisecond.

4.1 Predicting two second audio clips

The competition itself requires that the class of 5347 unlabeled samples is predicted. To make this work with this smaller model, the samples are also divided into 8 spectrograms. The result is 8 predictions for each two second sample, which we can then average together. Doing so can be a useful method, as this allows the predictions to be more robust against outliers. This is also proven by the increase in accuracy from 85% to 90%, and improvements in the PR-curve and confusion matrix seen in Figure 5.

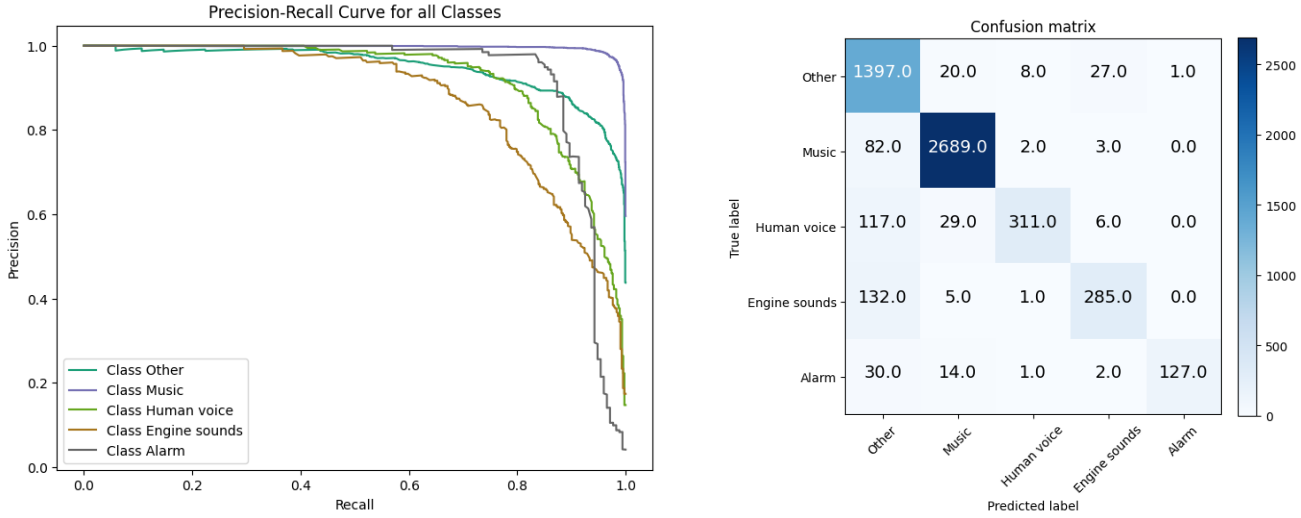

Figure 5: PR-curve and confusion matrix of the optimized CNN after averaging the predictions.

5 Discussion

5.1 Dataset

When splitting the two second clips up into 0.25 second clips we risk that some of those clips would not fit well with the labels. This is especially true for rhythmic sound signals who can have quiet periods, such as alarms or music (as can be seen in Figure 1). For these specific clips it could be beneficial to re-label them as background noise, or simply exclude them. Furthermore, the dataset is unbalanced with a high number of samples in the "Other" and "Music". The effects of this is also possible to see in the Figure 5, as the model performs best in those two categories.

5.2 Model optimization

To further reduce model size while also improving CPU and hardware accelerator latency, one can choose to optimize the model further. This does not just reduce latency, but also storage size and memory usage, while keeping the trade-off of reduced accuracy to a minimum.

A method for optimizing models are to perform quantization, which works by reducing the precision of the model parameters. These parameters are usually 32-bit floating point (as is ours), but can be reduced to 16-bit floating point or even 8-bit integer. Doing this, it is possible to reduce the model size by 75%, as well as speeding up the inference time with up to 3 or even 4 times [2].

6 Conclusion

In conclusion, a performance comparison of KNN, SVM and CNN for classifying; music, human voices, engine sounds, alarms, and others, to potentially remove noise in hearing aids was performed. The results showed that the CNN model outperformed both the KNN and SVM, in regards to validation accuracy, while still falling within the size limitations. All models tested had approximately an inference time between 1-2ms, which should allow for real time application. As well, the CNN shows promise in shorter periodic sound samples by splitting up the data into smaller chunks, allowing it to work with sound samples that are less then 1 second long.

7 Future Work

Introducing a method of dimensionality reduction to the solutions could potentially increase performance, while also reducing the size of the model. As can be seen from Figure 6 and 7, LDA partitions the data more coherently, and would therefore be interesting to implement in future developments of the solutions.

As well, rebalancing the dataset by adding more of non-music sounds could possibly increase accuracy of the systems. Without the need for recording new data, this could be achieved by data augmentation of the given data. Data can be augmented usefully by methods such as; noise injection, spectral augmentation etc. Noise injection is the act of artificially adding noise, which could be beneficial for the system as natural occurring sounds rarely are without noise.

References

[1] R. Thiruvengatanadhan, "Speech/music classification using mfcc and knn," *International Journal of Computational Intelligence Research*, vol. 13, no. 10, pp. 2449–2452, 2017.

[2] T. Flow. Model optimization. [Online]. Available: https://www.tensorflow.org/lite/performance/model_optimization

8 Appendix

Figure 6: PCA of dataset.

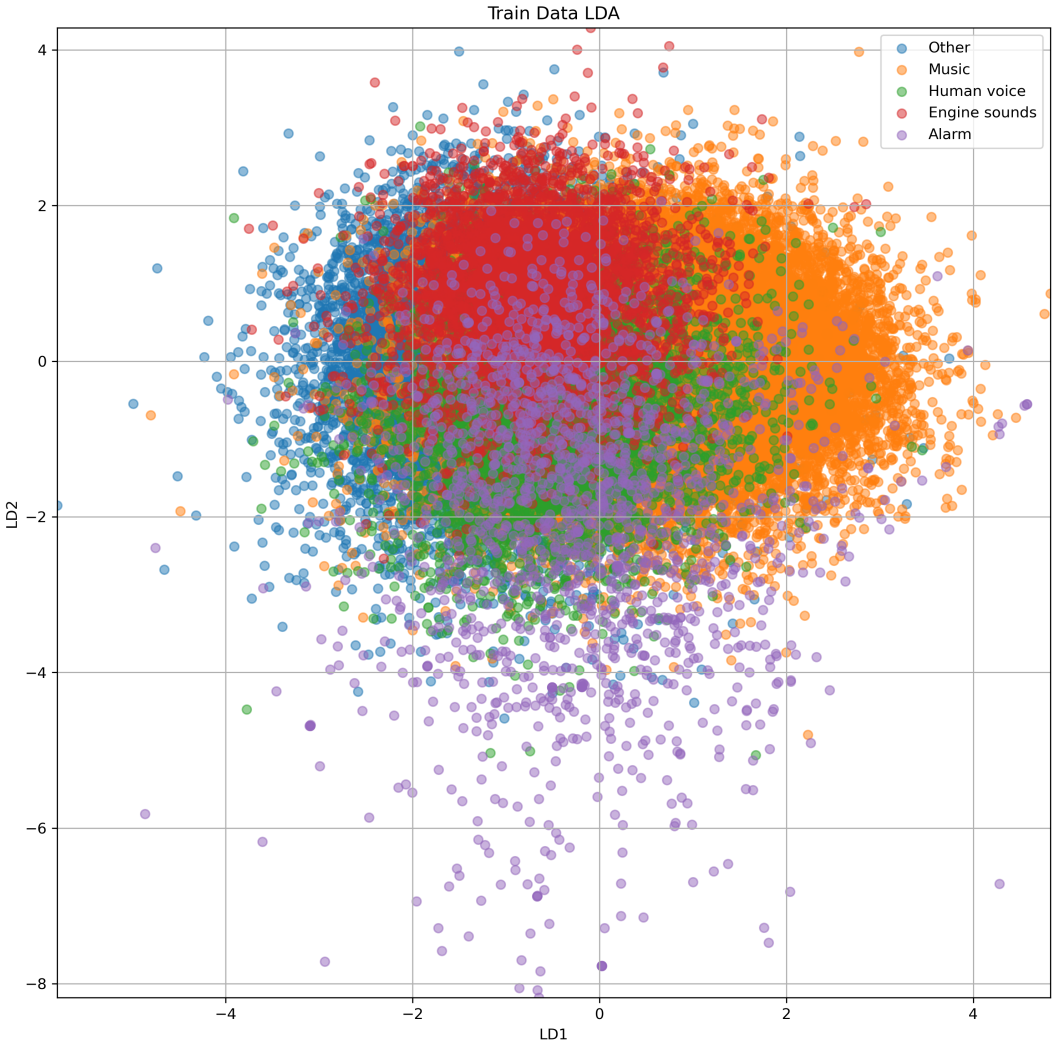

Figure 7: LDA of dataset.