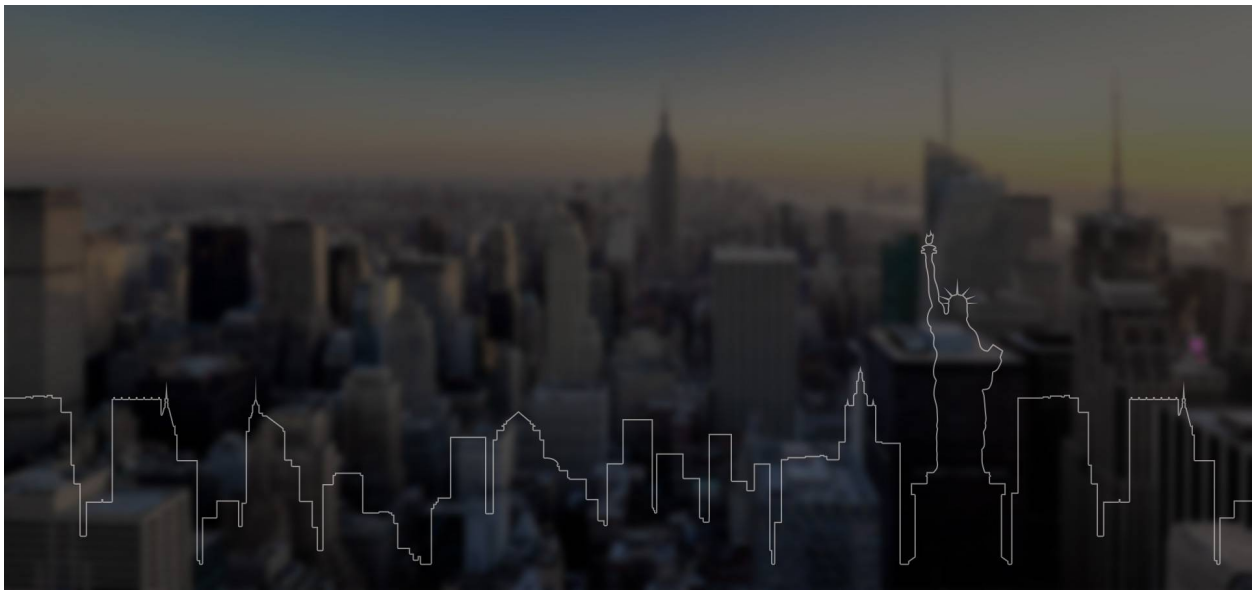


oticon

Oticon Audio Explorers challenge 2023

Sound Scene Classifier for Hearing Aids

Team Bihotz

Aiax Faura Vilalta
Irene Campillo Pereda

April 24, 2023

Contents

1 Introduction

The purpose of this project was designing a sound environment classifier using supervised learning techniques. The objective of this classifier is to automatically detect the nature of the sound environment to be able to adjust the settings of a hearing aid accordingly.

2 Data

The training dataset consisted of 52890 sound samples, each containing 32 frequency bands, of which each has 96 time frames. It was labeled with five different classes of sound environments with the data distribution from Figure 1.

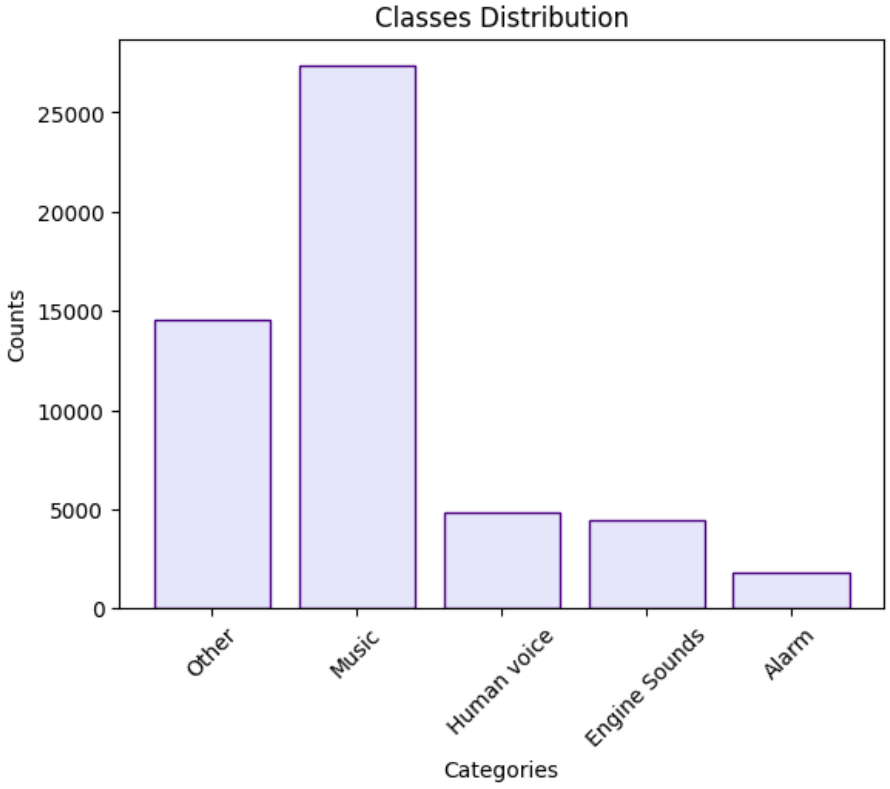


Figure 1: Training set distribution

The data was already preprocessed using the mel-scale, so no further preprocessing was necessary.

3 Evaluation

Before implementing the model is important to define the evaluation methods to see how the model performs for this specific problem. After considering different methods, the following

were considerate appropriate for this case.

3.1 Accuracy

Accuracy is perhaps the simplest metric one can imagine and is the proportion of sound scenes predictions the model got right. However, this metric can sometimes be misleading because high accuracy does not always mean good results. For example, for this data distribution, Music and Other account for 80% of the total data. So if the model predicted the data belonging to this classes correctly and the other ones wrong, would still get 80% accuracy which can be considered good. However, the other three classes are more critical. As stated in the challenge, it is important for the hearing-impaired listener to hear a vehicle approaching from behind in order navigate safely. And usually, the emergency of an alarm or the conversation coming from human voice will have priority over hearing music. Is because of this that the confusion matrix is considered a better evaluation for this project.

3.2 Confusion matrix

A confusion matrix shows the true positive, false positive, true negative, and false negative predictions made by the model. It helps to identify which classes the model is performing well on and which ones it is struggling with. To get a fast sense on how the model is performing for each category let's look at the F1-score.

3.3 F1-score

Precision measures the proportion of correctly predicted instances for a particular class, out of all instances that were predicted to belong to that class. While recall measures the proportion of correctly predicted instances for a particular class, out of all instances that actually belong to that class. For example, let's say there's 300 audio clips that belong to the class "alarm":

- Precision for "alarm": The classifier predicted that 50 audio clips are "alarm", but only 40 of them are actually "alarm". The remaining 10 are false positives. Therefore, the precision for "alarm" is $40/50 = 0.8$.
- Recall for "alarm": Out of the 300 audio clips that are actually "alarm", the classifier correctly predicted 40 of them. The remaining 260 are false negatives. Therefore, the recall for "alarm" is $40/300 = 0.133$.

Since for this case it is important both to trust that what the predictions say is true (precision), and that no audios are left out of the class predictions (recall), both metrics are useful. And F1-score combines recall and precision, so it can be estimated how the model performs for each class by looking at this.

4 Model

The neural network architecture that we defined for the sound environment classifier was designed to be computationally efficient and have a low memory footprint, while still being able to achieve high accuracy in classifying sound environments.

We chose a convolutional neural network (CNN) architecture because they are well suited for processing 2D inputs such as images and spectrograms. Convolutional layers can learn features at different scales by applying filters of different sizes to the input, and max pooling layers can reduce the spatial dimensionality of the feature maps while retaining the most important information.

4.1 Architecture

The architecture that we defined has two convolutional layers with relatively small filter sizes (3x3) and moderate numbers of filters (16 and 32). See figure 2. The small filter size allows the network to learn local features, while the moderate number of filters ensures that the network can learn more complex features without overfitting. The use of max pooling layers after each convolutional layer reduces the spatial dimensionality of the feature maps and helps to prevent overfitting.

The architecture also includes a dense layer with 64 units, which is followed by a softmax output layer with 5 units (one for each sound environment class). The dense layer allows the network to learn higher-level features that combine the local features learned by the convolutional layers.

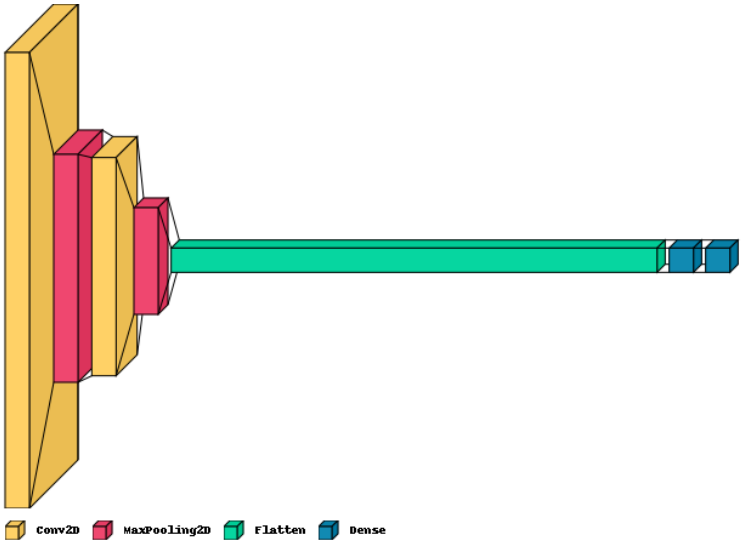


Figure 2: Model architecture

The model architecture has been defined based on the ones proposed by the papers [1], [2]

and [3], but adapting them to strike a good balance between model complexity and computational efficiency. This architecture has a total of 275,525 parameters, almost half of the maximum recommended 500,000.

All the implementations made for this project as well as the model weights can be found in this Github repository https://github.com/notaiax/Sound-scene-classifier-for-hearing-aids-ML

4.2 Training, validation and test sets distribution

Many distributions were tested to train the model. The one that gave best results was (Training set: 80%, Validation set: 20%).
However, the final model was trained on equally stratified data with (Training set: 80%, Validation set: 15%, Test set: 5%), so the model was able to see as much as possible of the data while we could still reassure that it generalizes on data never seen before. And since it's a large dataset, 5% is enough to validate. Even though this model peformed slightly worst than the first one, it was close enough and we can be sure that it doesn't overfit the training dataset.

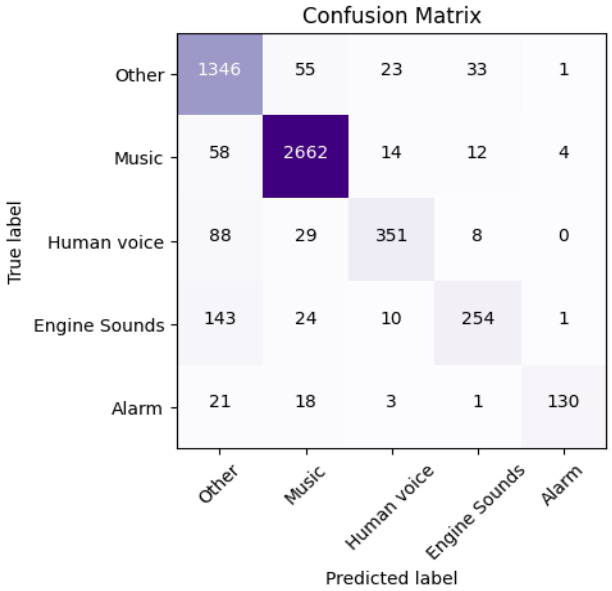
5 Results

We see from Figure 3 that the chosen model performs adequately on the validation set, as the diagonal from the confusion matrix has a stronger color, and the F1-score for all classes except from "Engine sounds" are higher than 80%. As we can see in the confusion matrix, most of the missclassified "Engine Labels" were predicted as "Others", this could be because as stated in the challenge, "Others" can cointain sounds from different classes, "Engine Sounds" in this case. This could be solved by removing the misleading sounds from the training dataset.

However, for the test set (see Figure 4) we see that the model don't generalize well in the 3 categories with less data, which happen to be the most important. So a possible improvement would be to balance the data with data augmentation.

Classification report:

	precision	recall	f1-score	support
0	0.81	0.92	0.86	1458
1	0.95	0.97	0.96	2750
2	0.88	0.74	0.80	476
3	0.82	0.59	0.69	432
4	0.96	0.75	0.84	173
accuracy			0.90	5289
macro avg	0.88	0.79	0.83	5289
weighted avg	0.90	0.90	0.89	5289

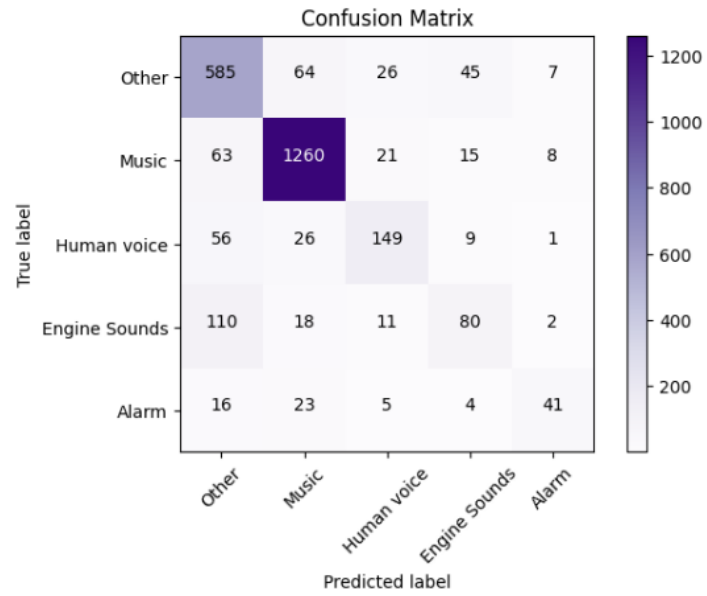


AUC score: 0.9810524088230453

Figure 3: Evaluation results on validation set

Classification report:

	precision	recall	f1-score	support
0	0.70	0.80	0.75	727
1	0.91	0.92	0.91	1367
2	0.70	0.62	0.66	241
3	0.52	0.36	0.43	221
4	0.69	0.46	0.55	89
accuracy			0.80	2645
macro avg	0.71	0.63	0.66	2645
weighted avg	0.79	0.80	0.79	2645



AUC score: 0.9329536749859617

Figure 4: Evaluation result on test set

6 Costs

The prediction time for each sample is: 1.96 milliseconds. Which should be more than enough for the purpose of this project.

The total training time for the model on Google Colab was 17min 23s with CPU and 1min 24s with GPU. During the training, CPU models have generally performed better and the final model was trained on CPU.

7 Future work

These are possible ways to improve the results that could not be tested and have been left as future work:

- It may be beneficial to collect more data or use data augmentation techniques to increase the size and diversity of the dataset. [5]

- The model could be improved by giving more importance and get better at detecting prioritary categories like alarms, engine sounds and human voice. This can be done with the following techniques:
  - Data augmentation to balance the data and improve the performance on the most important classes because they have less data.
  - Class weighting to assign higher weights to the prioritary categories during training to give them more importance. This can be done by setting the class_weight parameter in the Keras fit method.
  - The thresholds for the priority classes could be lowered so they have a higher chance of being predicted. However, this case should be studied and tested deeply before being applied.
- Use predictions explainability to see why the model makes the predictions it makes. This can be done with different tools such as Lime[6] and could help finding patterns in the data, see if the model predictions are reliable or just good by coincidence, and it also could help see why it missclassified "Engine Sounds" and possibly help cleaning the data.
- Finally, it would be important to test the model in real-world scenarios to validate its effectiveness in a hearing aid with realistic constraints on power consumption, memory, and processing power.

8 Conclusions

In conclusion, this project aimed to design a sound environment classifier using supervised learning techniques to automatically detect the nature of the sound environment and adjust hearing aid settings accordingly. A convolutional neural network (CNN) architecture was chosen for its ability to efficiently process 2D inputs and learn features at different scales. The model was trained on a dataset of 52,890 sound samples and achieved satisfactory results on the validation set, with most classes having an F1-score higher than 80%, except for "Engine Sounds."
However, the model's performance on the test set revealed that it did not generalize well for the three categories with less data, which are the most critical for the hearing-impaired listener. This issue could potentially be addressed by collecting more data, using data augmentation techniques, or adjusting the class weighting and thresholds during training.

Although is usually recommended to train CNN's on GPU, smaller CNN's sometimes perform better when trained on CPU and this was the case during this project.

The prediction time of 1.96 milliseconds per sound sample is deemed sufficient for the intended purpose of this project. Nonetheless, there is room for improvement, and future work may include refining the model to better detect priority categories, employing predictions

explainability to understand the model's decision-making process, and testing the model in real-world scenarios to assess its effectiveness in actual hearing aid devices.

Overall, this project represents a promising step towards the development of intelligent hearing aids capable of adapting to different sound environments, improving the quality of life for hearing-impaired individuals.

References

[1] Piczak, K. J. (2015). Environmental sound classification with convolutional neural networks. In *2015 IEEE International Workshop on Machine Learning for Signal Processing, Sept. 17-20, 2015, Boston, USA*.

[2] Salamon, J., & Bello, J. P. (2017). Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification. In *IEEE Signal Processing Letters, accepted November 2016*.

[3] Zeinali, H., Burget, L., Cernocky, J. (2018). Convolutional neural networks and x-vector embedding for DCASE2018 acoustic scene classification challenge. In *Detection and Classification of Acoustic Scenes and Events 2018, November 19-20, 2018, Surrey, UK*.

[4] Faura Vilalta A., Campillo Pereda I. *Sound Scene Classifier for Hearing Aids*, GitHub repository, https://github.com/notaiax/Sound-scene-classifier-for-hearing-aids-ML, 2023, Accessed: April 23, 2023.

[5] Ketan Doshi. *Audio Deep Learning Made Simple - Part 3: Data Preparation and Augmentation. Towards Data Science*, February 24, 2021. https://towardsdatascience.com/audio-deep-learning-made-simple-part-3-data-preparation-and-augmentation-24c6e1f6b52.

[6] Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin. *"Why Should I Trust You?": Explaining the Predictions of Any Classifier.https://dl.acm.org/doi/10.1145/2939672.2939778*. 2016