Audio Intent Detection Classification Problem

Ali Yassine
Politecnico di Torino
s312920
ali.yassine@studenti.polto.it

***Abstract*—In this report, I present a potential solution to the classification problem of the audio intent detection. To be more precise, the approach I suggested involves gathering the spectrum bands of each audio file and then determining various statistical features for each band. Based on statistical features, we inferred that two models were utilized to categorize new audio files. The solutions given by the proposed approach are superior than those provided by the constant baseline.**

I. Problem Overview

The proposed competition is a classification task using an audio intent dataset, which is a collection of audio recordings of English language commands from a variety of speakers from different nations and levels of proficiency. The idea of this competition is to properly identify each recording's stated command (action - object).

There are two different parts of the dataset:

- a development set of 9,854 recordings for which a label has been provided.
- an evaluation set of 1455 recordings.

Based on the development set, several considerations are possible. First of all, the data is completely unbalanced because almost 95% of respondents identified themselves as native English (United States) speakers. Second, samples from the recordings were mostly taken at two different frequencies 16000 Hz and 22050 Hz. Moreover, 96.95% of the recordings were sampled on 16000 Hz. Finally, recordings' lengths vary and are not consistent; in fact, we have 124 different lengths for recordings with 2.639s as the mean. Some outliers are present a with duration of 20s, which upon manual inspection appeared to be silence. I proposed a method for determining a common ground for all recordings and extract a specific amount of features to provide to my classification models because the submitted recordings had variable rate and duration.

The most interesting part of the evaluation set, however, was the fact that all recordings were made for individuals who self-identified as native English (United States) speakers and who regularly used the language at work or school.

In order to get a better understanding of our data, we can visually evaluate multiple signals in both the time and frequency domains. These two domains contain features for each signal. Figure 1 depicts one signal in the time domain. The recordings may contain silence, which could make up a sizeable portion of our recordings, in addition to voice instructions.

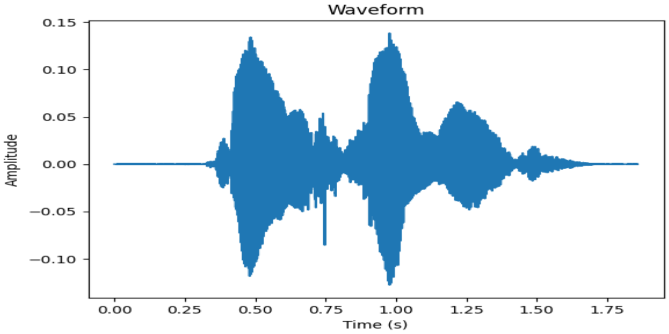

Fig. 1: Representation of a recording in the time domain.

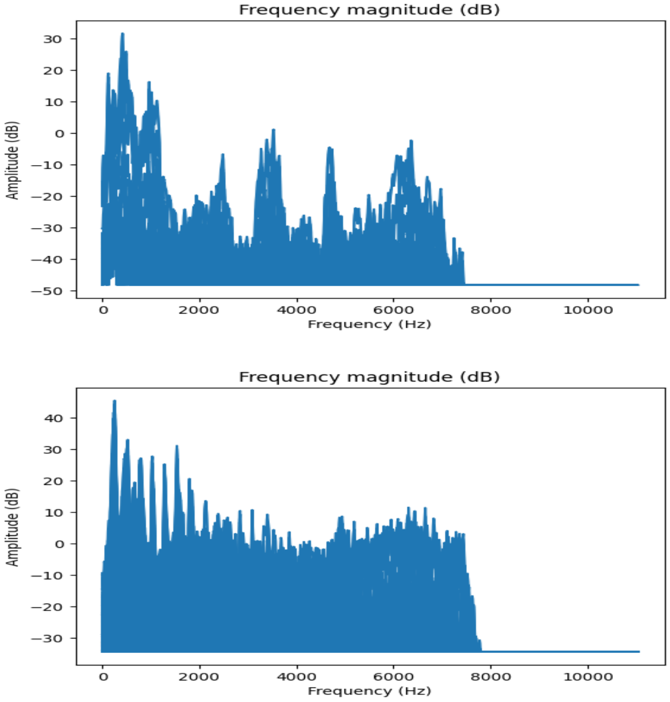

Fig. 2: Magnitude of two recordings of same class in dB

Regarding the frequency domain, in order to clearly illustrate the variation and uniqueness of the data at hand. Figures 2 shows the frequencies of two distinct recordings measured in decibels belonging to the same order given by a user.

II. Proposed Approach

A. Data preprocessing

As was previously mentioned, the data at hand is not clean; it is filled with unnecessary data that could be noise or silence. The following important steps were used to tackle this issue.

- Audio trimming: involves deleting unneeded information—in this case, silence—from the beginning and end of each audio signal.
- Noise reduction: involves eliminating unwanted audio sources by suppressing ancillary noises like fan noise.

A good illustration of the difference of the same recording before and after the trimming and noise reduction is shown in Figure 3 below.

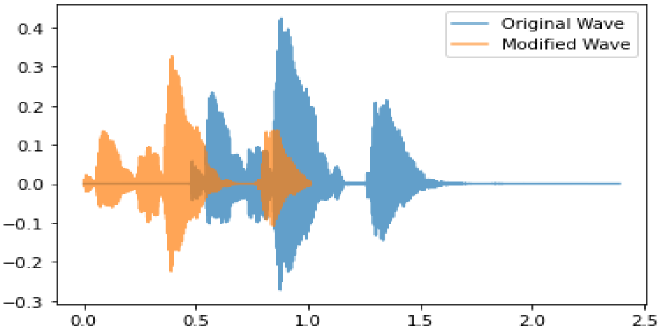

Fig. 3: Audio wave before and after modification

Audio feature extraction is feasible after altering the recordings that are provided. There is enough data in both the time and frequency domains for our models to use later, using Mel-frequency cepstral coefficients (MFCCs) [1], we may focus on both time and frequency domains at the same time. Figure 4 depicts the steps used for extracting MFCCs [2]. In terms of both the duration and the intensity of the MFCC coefficients, a MFCCs plot can be viewed as a matrix. Figure 5 provides a good illustration of a scaled MFCCs for better comprehension. Since our recordings were sampled at various rates, all of our audios were re-sampled to the same sample rate of 22050 Hz in order to have uniform feature extraction for all recordings. Afterward, we can extract a N x M matrix, where each row represents the MFCC coefficients for a particular frame of the audio signal and each column represents the number of MFCCs given as a parameter.

It is possible to combine valuable features after acquiring the matrix for each recording. Statistical data like mean, standard deviation, skew, min, max, etc. are collected in place of the entire data set to reduce the number of features for all recordings. To be more specific, since the number of columns is constant for all recordings unlike rows, the statistical methods previously discussed could be applied to the data contained in each column.

The number of MFCCs for each frame is the only hyperparameter used to derive MFCC features. Instead of retrieving all the data, this hyperparameter is typically defined to fix number of features to be extracted hence, reduce the dimensionality of our feature space.

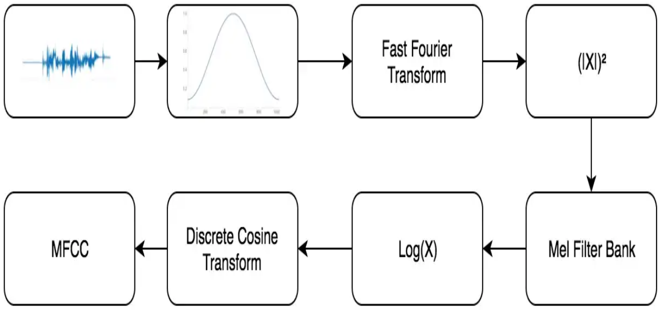

Fig. 4: MFCCs extraction algorithm [2]

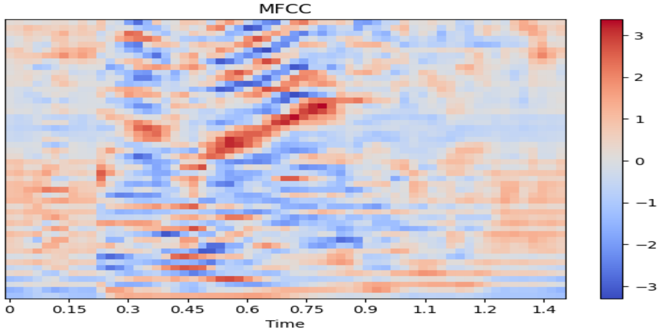

Fig. 5: Scaled MFCCs plot

Mel-spectrogram, a visual representation of the spectral power distribution of a sound wave with the frequency axis transformed from the linear scale to the Mel scale, is the secondary feature extracted. Every vertical column in a mel-spectrogram, which is typically retrieved as a 2D array, represents the energy distribution of a sound signal over various frequencies at a specific point in time. Additional statistical features of our recordings were collected in addition to the MFCCs, which are the primary features recovered for this problem. They include spectrum-related features such as:

- centroid
- contrast
- roll-off
- bandwidth

.

The zero crossing rate, which is frequently used for audio and speech-related issues, was also derived. It measures the rate of which a signal switches from being positive to zero to negative or the inverse.

B. Model selection

These algorithms have been examined:

- *SVM:* The supervised machine learning technique known as SVM, or support vector machine, can be used for both classification and regression tasks. It works by determining the optimal border between the various classes represented in the supplied data. When there are more features than records, the SVM approach is most useful. It can be useful for audio classification tasks, such as speech or music recognition [4].

- *Random Forest:* A supervised machine learning algorithm approach for regression and classification. It is made up of a collection of decision trees, each of which was trained using a distinct sample of the data and a randomly chosen set of features. The result is determined by averaging the predictions made by each tree. Audio classification tasks like speech or music recognition can be performed using random forests [5]. They can manage non-linear correlations between features and output in addition to handling a huge number of features.

The ideal combination of hyperparameters for both classifiers has been determined by a grid search, as described in the section below.

C. Hyperparameters tuning

There are two main sets of hyperparameters to be tuned:

- SVM model parameters
- Random Forest model parameters

After preprocessing our data and selecting suitable models a grid search algorithm, which is a technique that continuously searches through a manually defined subset of the targeted method's hyperparameter space, should be used to find the best combination of hyperparameters. Since accuracy is considered when selecting best hyperparameters, data from the development dataset is divided into train and test sets before the grid search method was executed based on the hyperparameters specified in Table 1. Figure 6 displays the accuracy of both models both before and after hyperparameter tuning.

Model	***Parameters***	***Values***
SVM	C	0.1, 1, 4, 8, 10, 50
	gamma	0.01, 0.1, 1
	kernel	rbf
Random Forest	n_estimators	10, 100, 1000
	max_depth	3, 5, 7
	min_samples_leaf	1, 2, 3
	criterion	gini, entropy

TABLE I: Hyperparameters considered

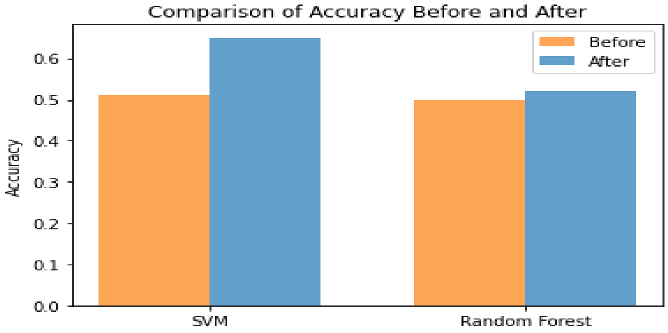

Fig. 6: Accuracy of both models before and after grid search

III. Results

Following tuning of the models hyperparameters of both SVM and random forest, as indicated before in Table 1. Accuracy improvements were mainly noted in SVM as shown in Figure 6. SVM, however, performed better than random forest. The best SVM hyperparameter combination was {C=4, kernel=rbf, gamma = 0.01}. For random forest, it was {criterion = gini, max depth = None, min samples leaf = 2, n estimators = 100"}. As was previously noted, SVM had a higher accuracy of 0.65 on the test set opposed to 0.52, for random forest. Furthermore, despite its lower performance, random forest managed to outperform the baseline. Both models achieved acceptable results.

Following the evaluation of both models, new models were fitted to all development datasets, and the models were then given evaluation data in order to classify them into classes. Public scores for the SVM and random forest were 0.647 and 0.5, respectively.

IV. Discussion

This paper proposes an audio intent classification approach to classify recordings into the correct class. Classification models were trained on slightly less than 10,000 recordings, and then testing was conducted on almost 1,000 recordings. Effectiveness is demonstrated by results that surpass the fixed baseline when utilizing MFCCs and other preprocessing techniques. But it should be noted that the suggested solution is only a beginning point, and that there is always room for improvement such as:

- Take into account additional features to extract from the recordings that are available.
- Apply convolutional neural networks as a model for classification [6].
- Use sophisticated filters to reduce data noise.
- Implement advanced feature selection algorithms.

The outcomes are encouraging, but they still need to be improved. However, it should be noted that the lack of further implementation was caused by limited resources, particularly those linked to computational performance.

References

[1] Salomons, Etto Havinga, Paul. (2015). A Survey on the Feasibility of Sound Classification on Wireless Sensor Nodes. Sensors. 15. 7462-7498. 10.3390/s150407462.

[2] M. A. Hossan, S. Memon and M. A. Gregory, "A novel approach for MFCC feature extraction," 2010 4th International Conference on Signal Processing and Communication Systems, Gold Coast, QLD, Australia, 2010, pp. 1-5, doi: 10.1109/ICSPCS.2010.5709752.

[3] Jolliffe Ian T. and Cadima Jorge 2016Principal component analysis: a review and recent developments Phil. Trans. R. Soc. A.3742015020220150202. Available:http://doi.org/10.1098/rsta.2015.0202

[4] Lu, L., Zhang, HJ. Li, S. Content-based audio classification and segmentation by using support vector machines. Multimedia Systems 8, 482–492 (2003). Available: https://doi.org/10.1007/s00530-002-0065-0

[5] L. Grama and C. Rusu, "Audio signal classification using Linear Predictive Coding and Random Forests," 2017 International Conference on Speech Technology and Human-Computer Dialogue (SpeD), Bucharest, Romania, 2017, pp. 1-9, doi: 10.1109/SPED.2017.7990431.

[6] S. Hershey et al., ”CNN architectures for large-scale audio classification,” 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 2017, pp. 131-135, doi: 10.1109/ICASSP.2017.7952132.