Audio Intent Detection using MFCC and Mel-spectrogram features with SVM

Stefano Chiartano
Politecnico di Torino
s313651
s313651@studenti.polito.it

Giuseppe Cavaleri
Politecnico di Torino
s314734
s314734@studenti.polito.it

***Abstract*—This report describes the approach adopted to solve the classification problem proposed. Due to making an efficient audio classification the mel-frequency cepstrum (MFC) suits with this case. From this representation MFCCs (Mel-frequency cepstral coefficients) can be extracted, allowing the construction of the features, used to train the model. Other features are considered using the mel-spectrogram of the signals, dividing them into chunks and averaging the values. This technique leads to good results in terms of the accuracy of the model, ensuring the overcoming of the minimum threshold.**

I. PROBLEM OVERVIEW

A. Dataset description

The dataset is formed by a collection of audio files in a WAV format. Each audio contains the statement of two words that identify an $action$ and an $object$. The purpose of the project is to train a model able to classify a group of audio files, determining what is the communicated command.

The complete dataset consists of 11.309 audio files split into two parts. The $development$ set contains 9854 records with several information about the audio files used to train the model. Each record represents audio and gives:

- Id: identifier of the audio;
- $path$: location of the WAV file;
- $SpeakerId$: identifier of the speaker;
- $action$: the action specified;
- $object$: the object specified;
- $Self-reported\ fluency\ level$: categorical attribute that explains the level of the speaker;
- $First\ language\ spoken$: the mother tongue of the speaker;
- $Current\ language\ used$: the language used in the work environment;
- $gender$: male or female;
- $ageRange$: interval of the age.

The other part is the $evaluation$ set that contains the rest of the audio files that must be classified, obviously, for these audio $action$ and $object$ are not provided. Doing a general analysis, the distribution of the data is not quite homogeneous. A large part of the audio is referred to the object $volume$, while the rest of the data is quite uniformly distributed as shown in Figure 1.

However, looking at the audio files, it is noticeable that the duration is not the same. In fact, neglecting the speaker's speaking speed, at the beginning and at the end of some recordings, we can identify some silence that does not give any information.

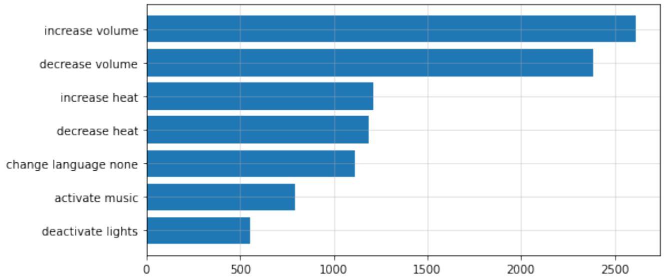

Fig. 1. Labels distribution

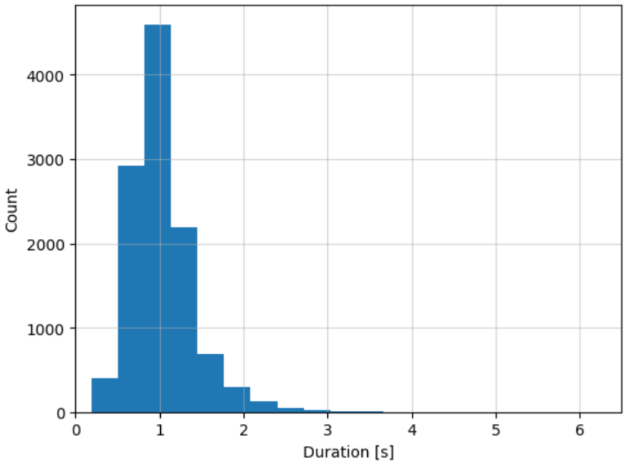

Fig. 2. Recording durations

II. PROPOSED APPROACH

A. Data preprocessing

The preprocessing stage of the audio recordings involved the elimination of silences throughout the duration of the audio and filtering of frequencies below 20dB. This resulted in a compact audio representation with a meaningful mean duration. The recordings were loaded with a constant sample rate of 16 KHz.

The result of this first part of data preprocessing is shown in Figure 3.

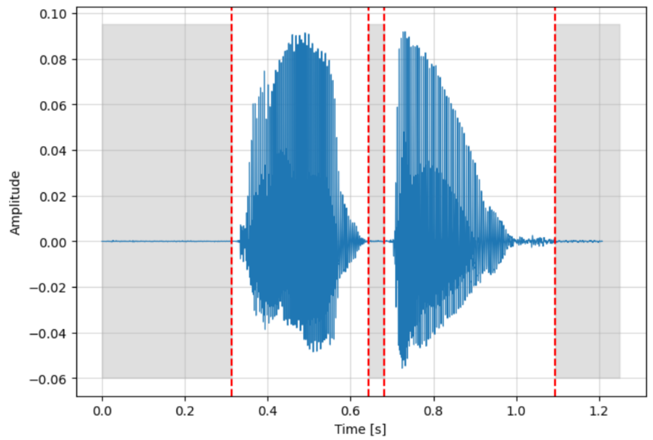

Fig. 3. Trimming of Silences from Audio 29

After this step, analyzing the dataset, we found two records which present excessive duration length. These records were identified as outliers and were completely removed from the development set. The amplitude is then normalized such that the values lie in the interval [-0.01, 0.01].

Principally, the features are extracted using two methods: mel-spectrogram and MFCC. A spectrogram is a visual depiction of a signal's frequency composition over time. The choice to adopt the Mel scale is due to the fact that the frequency bands are equally spaced, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal spectrum. It is related to Hertz by the following formula, where m represents Mels and f represents Hertz:

$$m = 2595 \log_{10}(1 + \frac{f}{700}) \qquad (1)$$

Each sample has a shape of 128 x 128, indicating 128 filter banks used and 128 time steps per clip. The Figure 4 displays the Mel frequency representation of an audio recording.

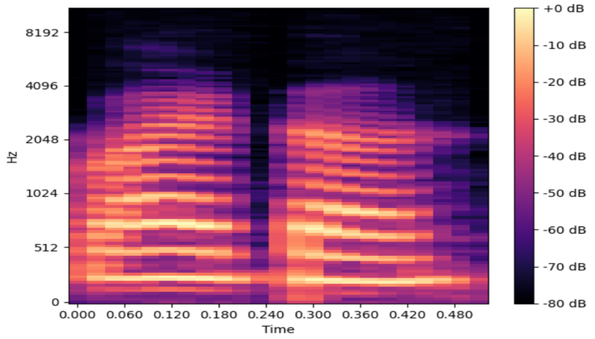

Fig. 4. Mel spectrogram of the Audio 29

In order to obtain a comprehensive view of the data and enhance the feature set, the second part of the process involves computing the MFCCs features. MFCC is an audio feature extraction technique which extracts parameters from the speech similar to ones that are used by humans for hearing speech, while at the same time, deemphasizing all other information. The $librosa$ API is utilized to extract n_f Mel-frequency cepstral coefficients (MFCCs) from each audio file. These features are then split into n_c chunks along the row axis. The mean value of the chunks is calculated to complete the process.

The same procedure is also performed on the first-order derivative (delta) of the n_f MFCCs features. This time, in addition to the mean, the standard deviation of each chunk is also extracted. (The standard deviation is calculated for this set of features as well.)

This process is shown in Figure 5, where $n_f = n_c = 8$.

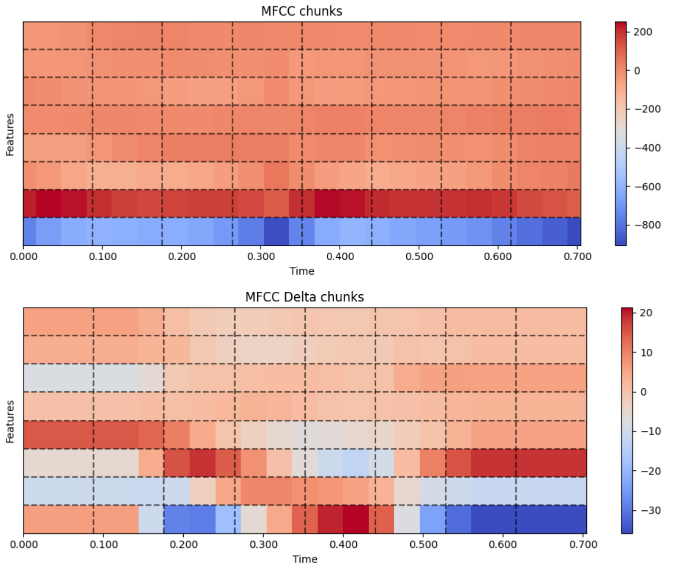

Fig. 5. Extraction of chunks respectively from MFCC and from MFCC Delta

After extracting all the features, a min-max normalization was applied to achieve a good fit between the model and the data representation.

Since the action-object label is categorical, a transformation was performed using an ordinal encoder to map each label to an integer. In the end, the inverse function was used to obtain the original labels after the transformation process.

Our models look to learn features from these two representations, and their architectures are described next.

B. Model Selection

In this study, two popular machine learning algorithms, Support Vector Machine (SVM) and Random Forest (RF), were tested as the classification models:

- $Random\ Forest$: RF is one of the most well-known ensemble algorithms that use decision trees as a base classifier. It selects a random sample with replacement

from the training set and trains the trees. Each tree is learned on a random set of features, typically formed by $\sqrt{p}$ features. Finally, the class is assigned by majority voting among the predictions.
- $Support\ Vector\ Machine\ Classifier$: SVM is a supervised learning algorithm that is widely used for classification tasks. It is a linear model that separates data points into categories by finding a hyperplane that optimally separates the categories. The hyperplane is selected such that it maximizes the margin between the data points from different categories. The training process analyzes audio training data to find an optimal way to classify audio frames into their respective classes.

C. Hyperparameters tuning

The tuning process involves adjusting two sets of hyperparameters:

- Feature Extraction parameters: n_f and n_c for MFCC, n_{row}, n_{col} for the spectrogram.
- Classification Model parameters: each of the two models we have studied have their own set of hyperparameters.

To optimize computation efficiency, a decision was made to set a uniform value for all four feature extraction parameters, designated as n

In order to find the best performing parameters for our data, we have run a grid search on the $development$ set, using the hyperparameters shown in Table 1.

	Hyperparmeters	**Values**
Preprocessing	n	4 → 22 step 2
RF	max estimators max depth	[100,300] [5,10,None]
SVM	C gamma	[4,20,100] [0.1, 0.01]

Table 1: Hyperparameter values

INSERIRE Grafico accuracy quando n cambia

III. Results

The accuracy of Random Forest and Support Vector Machine when n varies is shown in Figure 5.

As we can see from the graph, Support Vector Machine outperforms Random Forest.

The accuracy of the model increases until it reaches a peak in $n = 8$, after that it drops.

Using this grid search, we have found that the best configuration of parameters to our task is to set $n = 8$ and to use Support Vector Machine having $C = 4$, $gamma = .1$

In our program, we have noticed that combining MFCC features with the features extracted from the spectrogram, rather than using the MFCC features alone, improves the overall accuracy by a little margin as depicted in Figure 6, thus, both two methods can be considered.

Regarding other categorical features provided by the dataset, described in the first section, we have noticed that they do not improve model performance, thus we did not consider them in the final set of features.

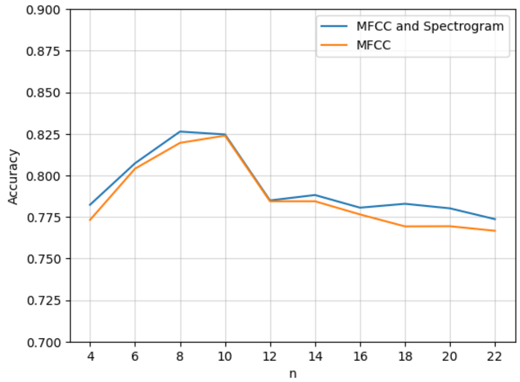

Fig. 6. Compared accuracy between

We decided to choose the version with both feature extraction methods, since for this model the best performance is achieved.

IV. Discussion

The models are trained using the optimal hyperparameters on the whole $development$ set and scored on the $evaluation$ dataset.

The public score we get using this technique is $accuracy = 0.941$, which, given the leaderboard, can be classified as a satisfactory result.

Nonetheless, some improvements could be made on this approach, in order to get better results:

- Use pre-trained models for speech recognition, such as $Google\ Speech\ Recognition$: these kind of models are pre-trained on much larger datasets, this allows to transcribe each audio into a phrase, which will be much easier to correlate to the action intended.
- Consider Convolutional Neural Networks: CNNs can be used as a feature extraction model to extract relevant features from audio signals. These features can then be used as input to another classifier, such as a recurrent neural network (RNN) or support vector machine (SVM), for the prediction of audio intents.
  Alternatively, CNNs can be trained as a standalone classification model to directly predict audio intents. This have been shown to produce good results for our task. [7]
- Explore more hyperparameters during the grid search, for example using different values for n_f and for n_c.

References

[1] V. Tiwari, ‘MFCC and its applications in speaker recognition’ International Journal on Emerging Technologies 1(1): 19-22(2010).

[2] MFCC features extraction [Online]. link:https://librosa.org/doc/main/generated/librosa.feature.mfcc.html

[3] MFCC delta [Online]. link:https://librosa.org/doc/main/generated/librosa.feature.delta.html

[4] B. Zhang, J. Leitner, S. Thornton, ’Audio Recognition using Mel Spectrograms and Convolution Neural Networks’, conference paper.

[5] L. Grama, C. Rusu, ’Audio Signal Classification Using Linear Predictive Coding and Random Forests’, link:https://ieeexplore.ieee.org/abstract/document/7990431

[6] L. Grama, L. Tuns, C. Rusu, ’On the Optimization of SVM Kernel Parameters for Improving Audio Classification Accuracy’, 14th International Conference on Engineering of Modern Electric Systems (EMES) 2017.

[7] D. Palaz, M. Magimai, R. Collobert, ’Analysis of CNN-based Speech Recognition System using Raw Speech as Input’, Idiap-RR-23-2015, June 2015.