Data Science Lab Winter Project

Politecnico di Torino

Mehrbod Nowrouz
Politecnico di Torino
s315925
s315925@studenti.polito.it

Mohammad Masoud Khalilian
Politecnico di Torino
s308102
s308102@studenti.polito.it

***Abstract*—In this report we aim to provide a possible solution for the intent recognition of the winter project dataset. Particularly, our solution consists of graphing the Mel Spectogram of the audio signals and splitting them into blocks of data. The summary statistics were extracted from these blocks and were processed and later used for training two classification models. The final results outperfoms a baseline goal set beforehand by the instructor of the course and thus it is considered to have reached a satisfactory result.**

I. Problem Overview

The project is a intent recognition problem on a dataset provided in winter project which consists of a collection of audio recordings of two word commands (such as ”Increase Volume”) from people of different age, gender and nationalities. The aim is to correctly classify these commands. The dataset provided is divided into two parts:

- **Development set** which contains 9854 recordings that have a specific id. The id connects each audio to a list of details regarding the command itself, the speaker and their linguistic characteristics.
- **Evaluation set** which contains 1455 recordings to be predicted.

The development set will be used to create a classification model which will detect the commands of the evaluation set.

Before starting the classification modeling, the development set was analyzed in order to reach a general idea of the data characteristics and how they should be dealt with later into the project.The dataset was found to be quite imbalanced in terms of data samples for each label. As you can see in the figure 1. The ”Increasevolume” label has over 2500 samples whereas the sampe amount of alternate labels such as ”deactivelights” has not even gone past the 1000-count threshold. This phenomena which is often referred to as imbalanced classification is a common issue which if not dealt with accordingly would lead to biased outcomes in several ways:

- **Bias towards majority class:** When the majority class is significantly larger than the minority class, a model may end up predicting the majority class more often, even if the data is not actually representative of the true distribution.
- **Poor performance on minority class:** Models may have difficulty in correctly classifying instances of the minority class, leading to a high false negative rate.
- **Metric Misrepresentation:** Common evaluation metrics such as accuracy, precision, and recall can be misleading in imbalanced classification problems, as they do not take into account the imbalance in class distribution.

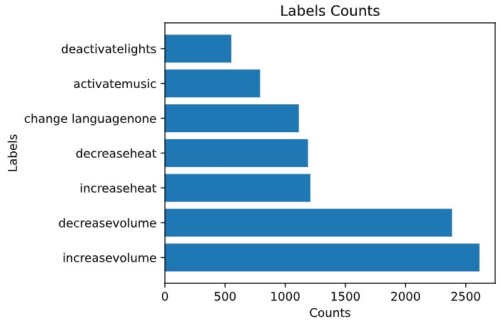

Fig. 1: Labels frequency

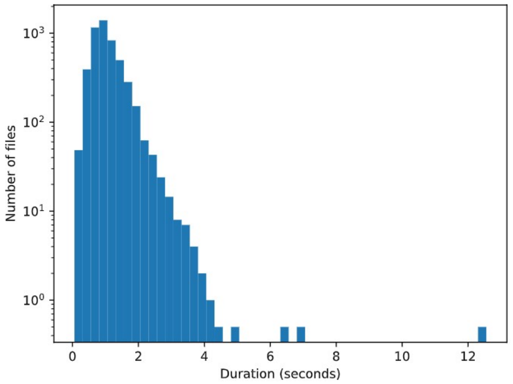

Fig. 2: Labels frequency

Furthermore, the sound signals durations are varied and as visible in figure 2 they are mostly distributed between about 0 to 5. There are some audios that tend to have a significantly higher duration compared to the common trend (6 to 13 seconds). These signals were manually inspected, and it was found out that they contained a considerable amount of silence contributing to the high durations. Since there are varying lengths in the audio samples, we need to present a solution that would be flexible towards audio lengths and yet manages to be accurate despite the variety.

It is also notable that during manual inspection of the audio files, some audios were found that did not seem to be audible. These samples contained no useful info and would mislead the future algorithms and consequently, the model provided should be able to address these audios.

While these audios are in linear scale, since human perception of sounds is logarithmic [1], we transform the original linear audio samples to logarithmic ones (dB scale) and use them instead in the training our model.

II. Proposed Approach

A. *Data preprocessing*

Audio time and frequency domains are rich with information and such information can be the key to classification of the audios into their desired labels. These audio features were sampled in a sampling rate of 22050 Hz which means that the number of samples per second in the audio signals are reduced or increased to match the target sample rate. These samples were then used to create a mel spectogram of each audio. However, the result is a Mel Spectogram of the linear audios. As previously explained, we desire a logarithmic version of inputs into our models. Therefore, the resulting Mel Spectogram is transformed into a logarithmic one. The figure 3 resembles the result of the above steps. However, as visible in the example figure, some of the Mel Spectograms tended to include areas that lacked information regarding our audios as a result of occasional long silences before and after the speaker's commands in our dataset. These areas not only do not contribute to better understanding of our codes, but they also lead to miss classifications since the same commands might have different amount of silences in the audio samples. Also, the audios that were inaudible tended to have a mostly black spectogram and contained next to no information. Therefore, all the audio files were trimmed to be stripped of such absence of information and the ones that were inaudible were all removed from the dataset using a threshold limit for the energy of a voice. Not to be unsaid that an algorithm was programmed that classified the audio files based on age and first language spoken based on which some specific audio files were removed from the training set which did in turn boost the accuracy by a small portion.

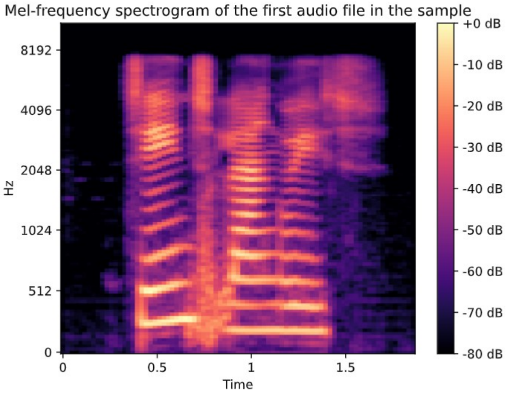

Fig. 3: Mel Spectogram before Trimming

The resulting Mel Spectograms of the same audio signal is illustrated in figure 4. Eventually, in order to extract the visible features on the spectogram in a way that computer can recognize it, the spectogram was divided by a N x N grid into N^2 bins of the same size. The mean and standard deviation of each bin was calculated. Each of these calculations were treated as a feature of our audio signal and was put inside a matrix of shape N rows and N columns with respect to the bin's position in the grid. This approach comes with major benefits such as:

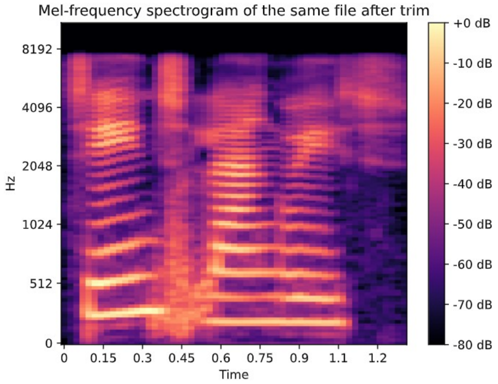

Fig. 4: Mel Spectogram after trimming

- **Flexibility in the number of features:** The number of bins and consequently the number of features can be easily changed into the desired number to obtain better results.
- **Generation of a uniform number of features:** Since the number of bins is independent of the audio's length, this approach can help us generate the same amount of features regardless of the audio length.

Selecting bins in a way that the numbers of bins are not dependent on audio length will result in each bin including a variable number of features in them. In other words, we are ”squeezing” or ”stretching” the audio signals in time. We treat N as a hyperparameter to find the value which produces the best output in the end.

B. *Model selection*

The following algorithms have been used:

- **Random Forest:** This algorithm functions through creating multiple decision trees and decides on the final classification result by choosing the output that is elected by a majority voting among the decision trees. It is notable to mention that random forests classify audio signals through working on features one at a time which means the input data do not need normalization and While this approach is more accurate compared to decision trees, it sacrifices interpretability to some degree. The performance of decision trees scales up with the number of estimators until a certain point. [2] and we chose this algorithm, since it is known to be performing well on audio signal classification. [3]
- **SVM:** The main idea behind SVMs is to find the best boundary, called a hyperplane, that separates the data into classes in such a way that the margin between the classes is maximized. This approach is also known to be producing rather good results on audio classification problems [4] and therefore, it was used during this project. They usually require normalization.

Both models above were trained and executed. However, the hyperparameter used for the first runs were not final and required tuning which is explained in further details below.

C. *hyper parameter tuning*

There are 2 main hyperparameter sets present in our models consisting of:

- **N:** The number of vertical and horizontal lines that separated our Mel Spectogram into N^2 bins is yet to be tuned.
- **SVM and Random Forest Parameters:** Each of these models require configuration in parameters such as max-depth, evaluation criterion (Random Forest), C and gamma (SVM) parameters.
  To perform the required tuning, we chose the grid search approach for both set of hyperparameters. To find the best values among our grid search parameters, we separated our dataset into 80/20 percent training set/test set splits and tried all the parameters in the grid search to discover the best values for each of the hyperparameters. The details regarding the values assigned to each of these hyperparameters are visible in the table 1.

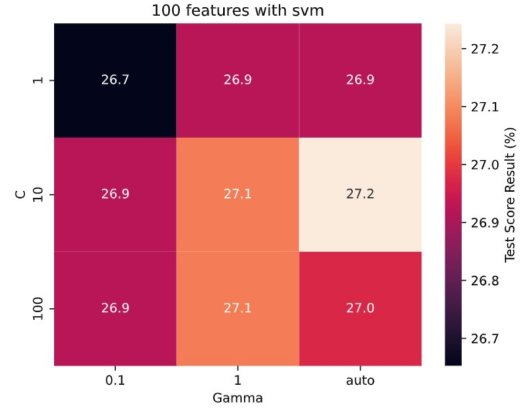

Fig. 5: SVM Accuracies

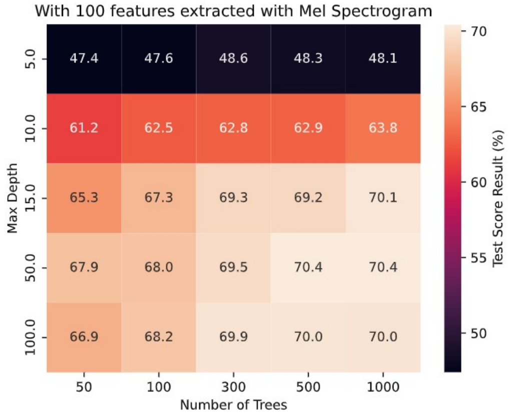

Fig. 6: Random Forest Accuracies

III. Results

As visible in figure 6, despite the attempts on tuning SVM parameters not only were the results comparable to Random Forest's but they also under performed and barely reached the baseline declared for the project. Random Forest on the other hand did show signs of improvement by hyperparameter tunings and managed to reach an accuracy of 70.4 percent compared to initial accuracies of nearly 50 percent. The public score of the model in the leaderboards managed to perform even better and achieved an accuracy score of 75 percent. These results as partially visible in figure 6 were produced with the configuration of: [N = 10, criterion: entropy, max_depth: 50, n_estimators: 1000]. The overall performance of random forest managed to surpass the baseline by approximately 40 percent and thus the model is considered successful, and the results are satisfactory. The configurations used for the random forest are as follows:

Model	Parameters	Values
Preprocessing	N	[5,7,10]
Random Forest	n_estimators max_depth criterion	[50, 100, 300, 500, 100] [5, 10, 15, 50, 100, None][’gini’, ’entropy’]
SVM	C gamma	[1, 10, 100] [0.1, 1, ’auto’]

TABLE I: Hyperparameters considered

IV. Discussion

The presented approach outperformed the naive baseline set in the leaderboards. It made good use of frequency and time domain features and managed to obtain nearly the best configuration for the task at hand. However, we managed to achieve a score of 75 percent and we observed in the leaderboards that results of over 95 percent were also possible. Many attempts were made to take into account the remaining possible features in the datasets in order to improve our final results, although they didn't manage to surpass the current results. In case of a more extended deadline, we believe it might have been possible to achieve a better accuracy through the usage of approaches stated below, ordered based on their importance and what we believe would have impacted the results the most:

1) **Data Preprocessing:** Countless methods regarding data preprocessing such as usage of MFCC (which is often used in voice recognition[5] were made but none of them managed to surpass the current method. Nonetheless, we believe the key towards better performance in this project lies within the data preprocessing. There were many aspects of an audio signal that were not included as a feature. A rather effective solution that could potentially improve the outcome was balancing the sample distribution of labels through oversampling the minority class, under sampling the majority class, or using more advanced techniques like Synthetic Minority Over-sampling Technique (SMOTE).
2) **Neural Networks:** CNN's have proven to be quite promising in classification of audios. Their usage would probably have led to better results.
3) An interesting yet complex approach towards preprocessing of the audios could be seperation of the audio into two parts of verb and object through the silence between the utterance of the two words and separate classification of the verbs and objects in the audios.
4) **SVMs:** SVMs are known to be performing well on audio signals despite the results we achieved. It is likely that there is a solution in making them produce better results.

In the end, the current results are already very promising and have managed to reach the demanded expectations. This problem despite its challenges managed to help us better grasp the fundamentals of audio classification and learn the theoretical lessons in a more practical way and further boosted our thirst in diving deeper into the world of Data Science.

References

[1] E. B. Goldstein, Sensation and perception (3rd ed.). Wadsworth/Thomson Learning, 1989.

[2] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas, "How many trees in a random forest?" in International workshop on machine learning and data mining in pattern recognition. Springer, 2012, pp. 154–168.

[3] . Saki, A. Sehgal, I. Panahi, and N. Kehtarnavaz, "Smartphone-based real-time classification of noise signals using subband features and random forest classifier," in 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2016, pp. 2204–2208.

[4] P. Dhanalakshmi, S. Palanivel, and V. Ramalingam, "Classification of audio signals using svm and rbfnn," Expert Systems with Applications, vol. 36, no. 3, Part 2, pp. 6069 – 6075, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417408004004

[5] P. Dhanalakshmi, S. Palanivel, and V. Ramalingam, "Classification of audio signals using svm and rbfnn," Expert Systems with Applications, vol. 36, no. 3, Part 2, pp. 6069 – 6075, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417408004004