Data Science Lab: Lab 7 Report

Nunzio Licalzi s344860
DAUIN - Dipartimento di AUtomatica e INformatica
Politecnico di Torino
Torino, Italy
s344860@studenti.polito.it

***Abstract*—This report describes the solution adopted to solve the free spoken digit classification problem. In particular, the approach revolves around extracting the spectrogram of each signal, the compression of said spectrogram using two different methods, and classifying the signal using a mix of Random Forest, SVM and KNN classifiers.**

I. Problem Overview

The free spoken digit problem is a well-know classification task. It involves classifying audio recordings of varying lengths. In each audio, a different speaker (with different sex, age and accent) pronounces a digit (from 0 to 9).

The goal is to correctly identify and classify an evaluation set of 500 audio samples, starting from a training set of 1500 audio sample.

The main challenge of this task is related to the variation of content in each audio file (background noise, duration, and speaker) and the relatively small training set.

II. Data Exploration

The first step was to examine the provided sample set to better understand the data we are dealing with.

First, all the audios were sampled at 8KHz, eliminating the need to address issues related to varying sample rates. Additionally, the problem is well-balanced, with exactly 150 files per class.

As previously mentioned, the main challenge lies in the differing lengths of the audio recordings. The lengths are normally distributed with mean $\mu = 0.4$seconds and standard deviation $\sigma = 0.128$seconds. Unfortunately, there are several outliers, the most notable of which is approximately 2.2 seconds long, as shown in Figure 1.

This poses an issue, as pre-processing is required to deal with the digits being spoken at different times.

It is crucial to remember that we are ultimately working with signals, which leads to analysis both in the time domain and frequency domain.

Transitioning to the frequency domain yields promising results, but the key task is to ensure a fixed number of features for model training.

III. Proposed Approach

The final approach I chose involves classifying the signals based on features extracted from a synthesized spectrogram.

Initially, two other approaches were evaluated: energy and power of the signal, and correlation among signals.

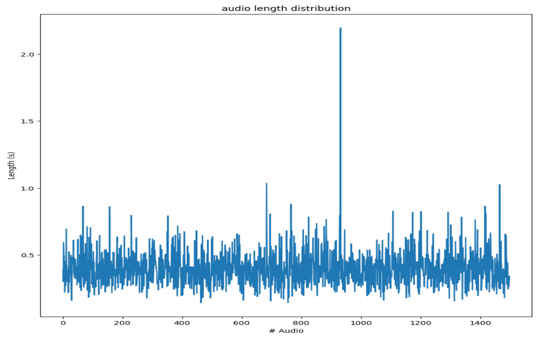

Fig. 1. audio length distribution

The first approach did not provide a satisfactory F1 score, while the second approach was unsuitable because correlation works well only when comparing the same signal shifted in time by a time constant τ such that $x(t) = x(t+\tau), \forall t \geq 0$, although it generally performs well with noisy data [1].

A. Data Pre-processing

Each file is loaded and converted into its corresponding spectrogram, which is saved as a two-dimensional matrix in $\mathbb{R}^{N_i, M_i}$, where N_i is the number of frequency bins for the i-th signal and M_i is the number of time bins for the same signal. Note that the amplitude of the signal is converted to decibels (dB) because human hearing uses a logarithmic scale rather than a linear one [2].

The issue of varying signal lengths persists, as each signal generates a matrix with a different number of rows and columns. Additionally, the large number of values stored in this manner cannot be directly used for classification.

To address these issues, the following strategy was adopted: each matrix is reduced in dimensionality to a fixed size with N rows and M columns. The parameters N and M are chosen as fixed values for all signals.

To achieve this, the matrix is grouped in such a way that the dimensionality becomes $N \times M$, and for each group of data, the mean and standard deviation are computed and stored in two separate matrices each of which has a dimension $N \times M$. This ensures that each signal has the same number of features, which can be used for classification.

Finally, the mean and standard deviation matrices are linearized in a row-major order and the obtained values are used as features for classification.

B. Model Selection

Before selecting the classification model(s), it is important to note that the pre-processing parameters N and M are hyperparameters and, as such, they must be accurately tuned, just like the model itself. The following models were evaluated:

- **Decision Tree**: The first model evaluated was the decision tree, mainly because it is one of the most interpretable models. However, as expected (will be discussed later), it was not sufficient to be a viable solution.
- **SVM**: This model is known to perform well with audio signals [3].
- **KNN**: This model showed promise due to the solid results obtained from previous usage in audio classification [3].
- **Random Forest**: This model uses a set of decision trees, trained independently on different portions of the data, and classifies the final result using majority voting. It is almost as fast as a decision tree but less interpretable.

All the models were fine-tuned, yielding varing results.

C. Hyper-parameters Tuning

There are several hyper parameters to be tuned:

- Pre-processing parameters
  - Number of rows N
  - Number of columns M
- Models parameter related to
  - SVM
  - Random forest
  - KNN
  - Decision tree

The following assumption was made for N and M $N \leq min(N_i), \forall i$ and $M \leq min(M_i), \forall i$.
A comprehensive table of the tested hyper parameters is shown in Table I.

Model	Parameter	Tested	Chosen
pre-processing	N	[20, 80] with a 3 step	26
	M	[3, 6]	4
Random forest	*criterion*	gini, log_loss, entropy	entropy
	n_estimators	[100, 800) with 100 as step	200
SVM	*degree*	[3, 10)	3
	kernel	linear, poly, rbf, sigmoid	linear
	probability	True, False	True
Decision tree	criterion	log_loss, entropy, gini	log_loss
	splitter	best, random	random
	max_depth	None, 4, 8, 10	10
KNN	*n_neighbors*	5, 10, 20, 30	30
	algorithm	auto, ball_tree, kd_tree, brute	auto
	p	1,2,3	1
	weights	uniform, distance	distance

TABLE I
HYPERPARAMETER TUNING

NOTE: The F1 score testing was performed using each individual model, while the configuration for N and M was tested with all the models listed (except for the decision tree, for reasons that I will explain shortly).

Model	F1 score
SVM	0,9722
Random forest	0,9643
KNN	0,9602
Decision Tree	0,7069

TABLE II
BEST F1 SCORE FOR EACH MODEL

This table II shows the best F1 score achieved for each model. Given the achieved scores, the first three models were selected, while the decision tree was discarded and not used for classification due to poor F1 score. The final classifier is created by combining the top three performing classifiers into a single model.

How does the final model behave?

1) The three models are trained independently using the same dataset.
2) Each model then predicts the class for each data point independently.
3) A weighted average is computed, using the F1 scores achieved during the validation phase as the weights, and the final class for the audio signal is determined via majority voting.

IV. DISCUSSING THE RESULTS

The final score obtained on a test set (80/20 split) was 0.96, while the final public score, after training the model on the entire dataset, was 0.98. The following confusion matrix was obtained, as shown in Figure 2.

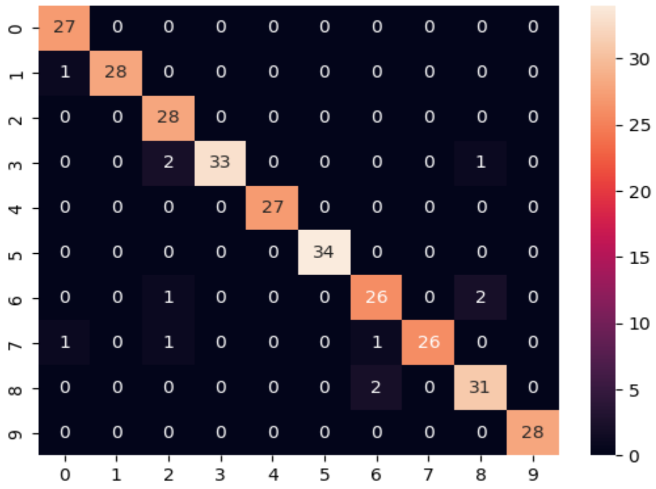

Fig. 2. Confusion matrix

After conducting thorough research online, I found additional data similar to the ones used for training [4].
When these new data were added to the original 1500 samples, the model achieved a final F1 score of 1.00. (I suspect that the test set for the online evaluation included some of the data in the new batch).
The final results obtained, even without the new data, are very promising, and there is little else that could be improved. Perhaps removing noise from the signals could provide a small

improvement. Additionally, I do not foresee any issues with future testing using undisclosed data.

REFERENCES

[1] L. Costa, Comparing Cross Correlation-Based Similarities, available here.

[2] C. Stangor, J. Walinga, Introduction to Psychology – 1st Canadian Edition, Chapter 5.3 Hearing

[3] Environmental Sound Classification using Hybrid SVM/KNN Classifier and MPEG-7 Audio Low-Level Descriptor, available here

[4] The dataset is publicly available here