Intent Recognition using Vocal Recordings: Neural Network Classification based on Raw Audio Signals

Michele Ferrari [*21118273*]

06/11/2021

Table of Contents

Introduction

The task is to classify voice recordings into three different classes:

- Approval
- Prohibition
- Attention

Data come from two datasets: *Kismet* and *BabyYears*.

1. Data Import

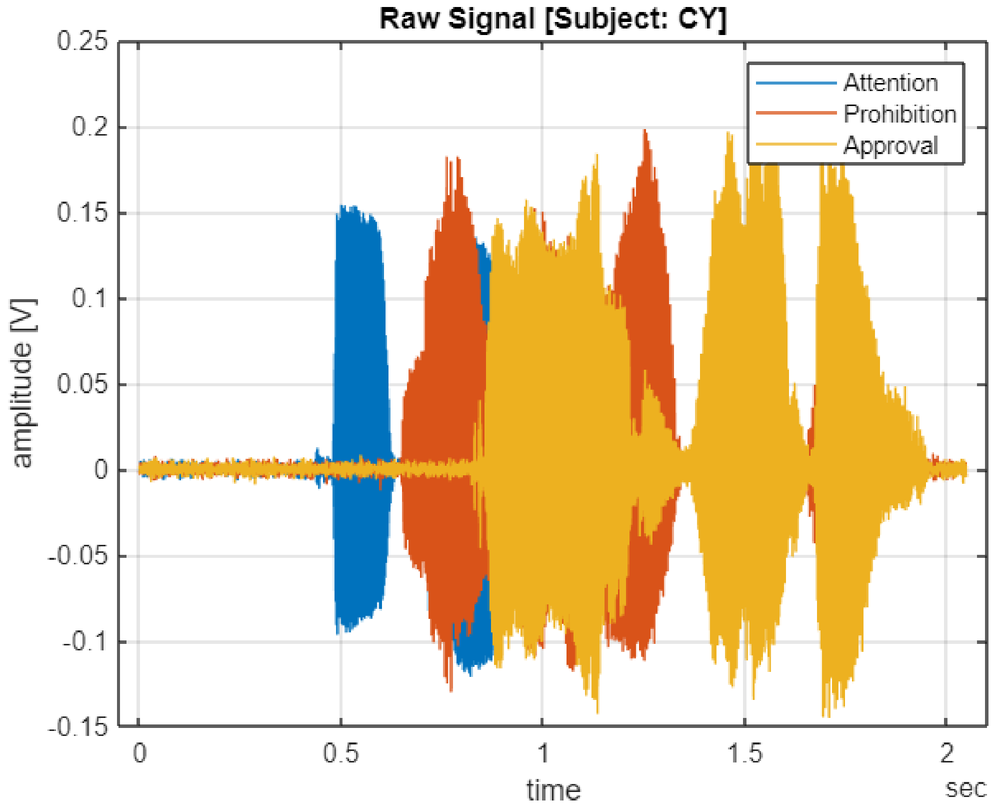
Automatically import all `.wav` files and organize them in proper data structures.

2. Data Visualization

Use Kismet Dataset and subject CY as an example.

2.1 Display signal in the time domain

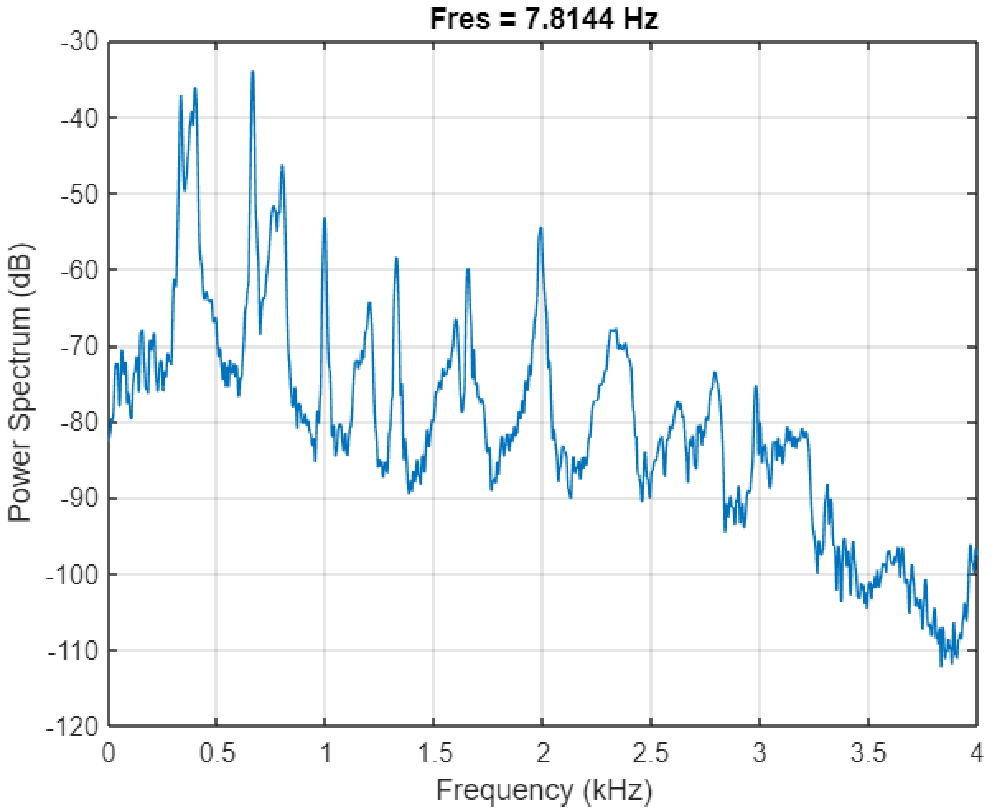
Example of the content of `.wav` files. It is the raw signal, as registered by the microphone.

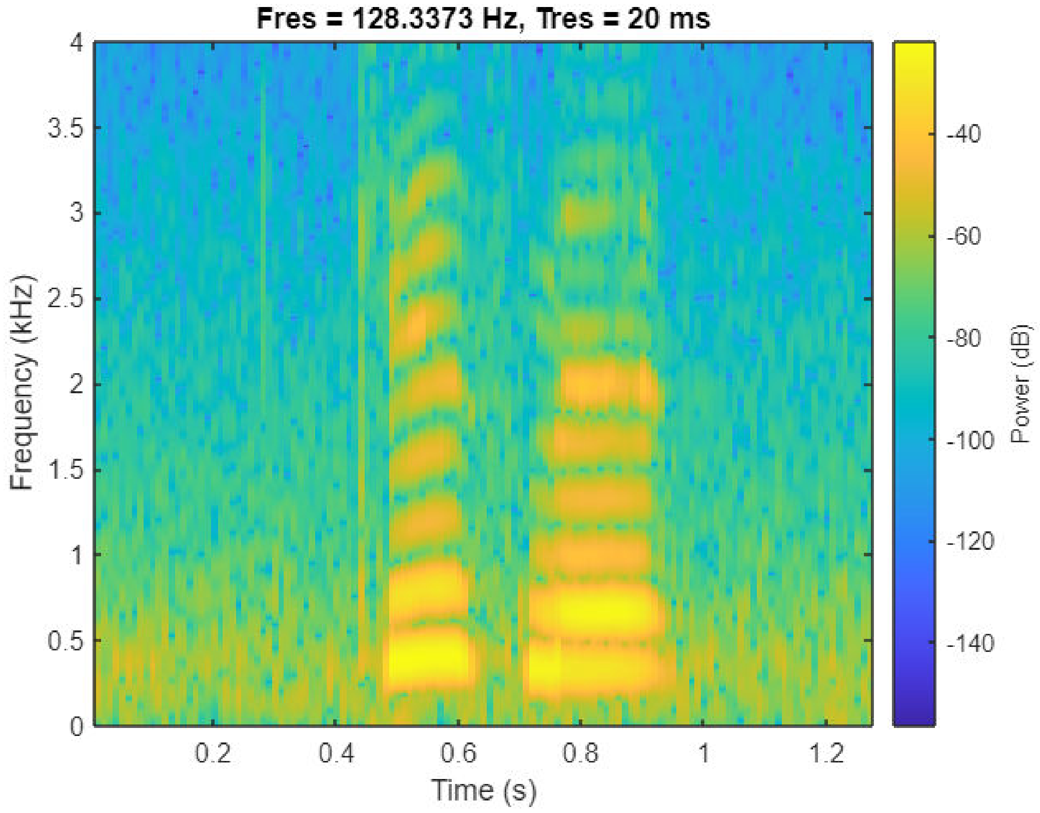


2.2 Basic spectral information about the wav signal

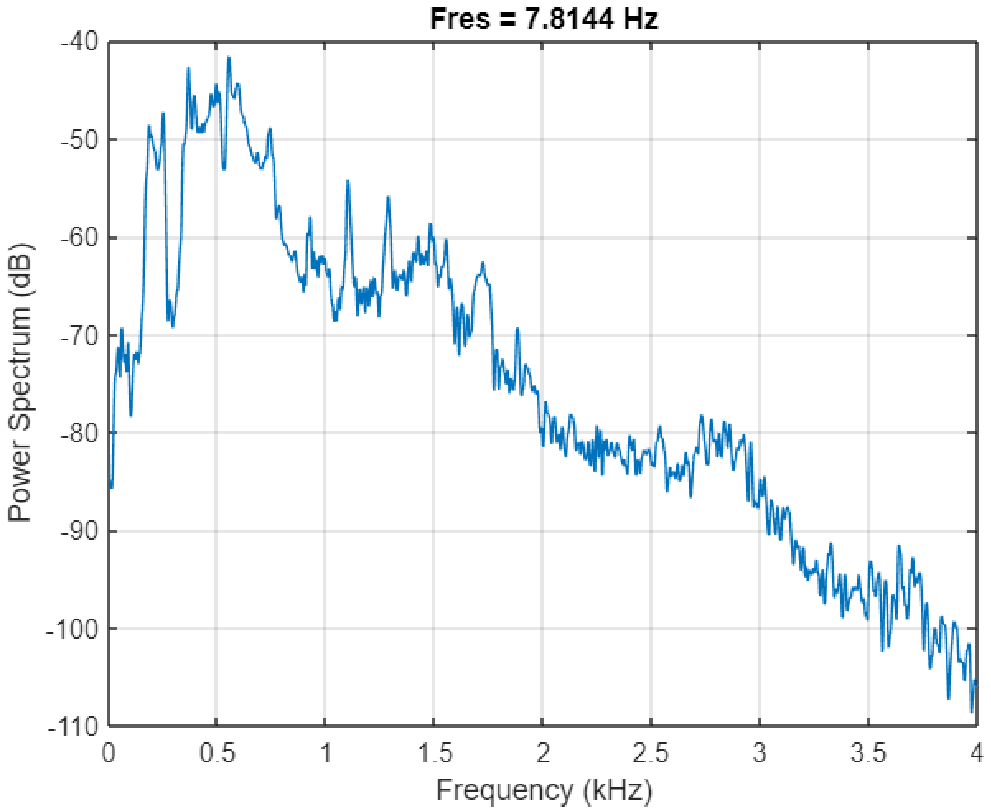
Here the Power Spectral Density and the spectrogram of the signals are displayed, in order to show the spectral differences among different classes. These informations will be used to classify signals.

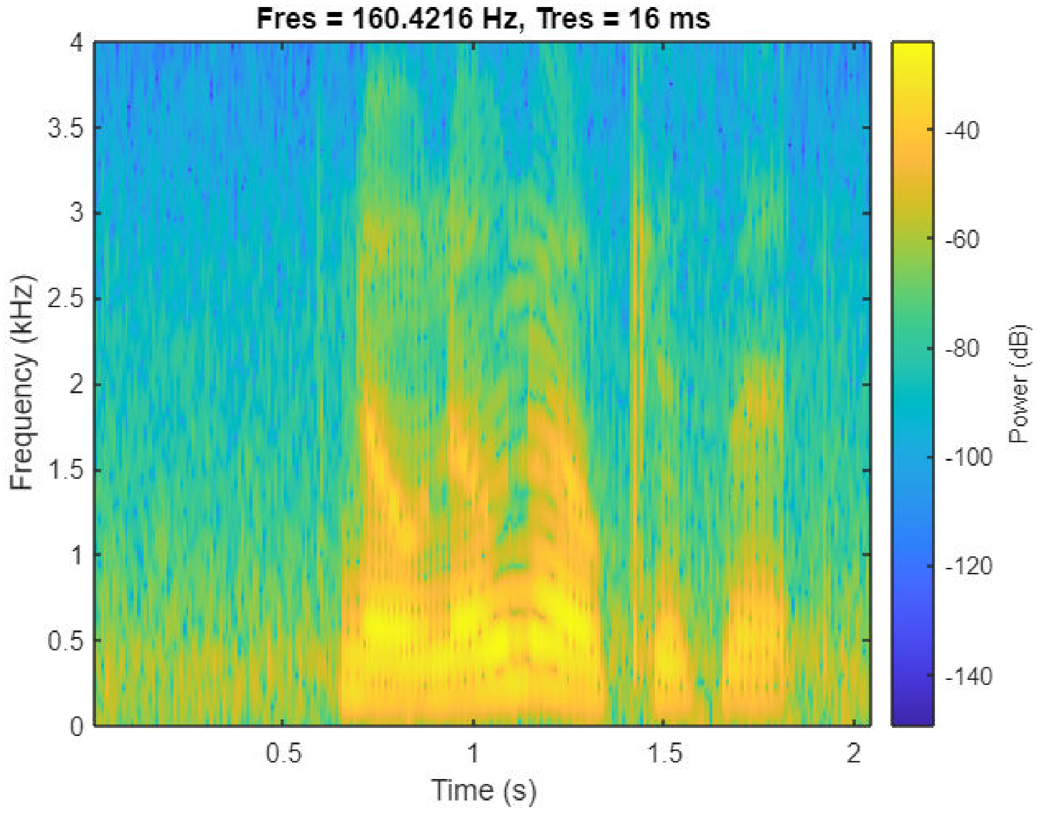
Attention



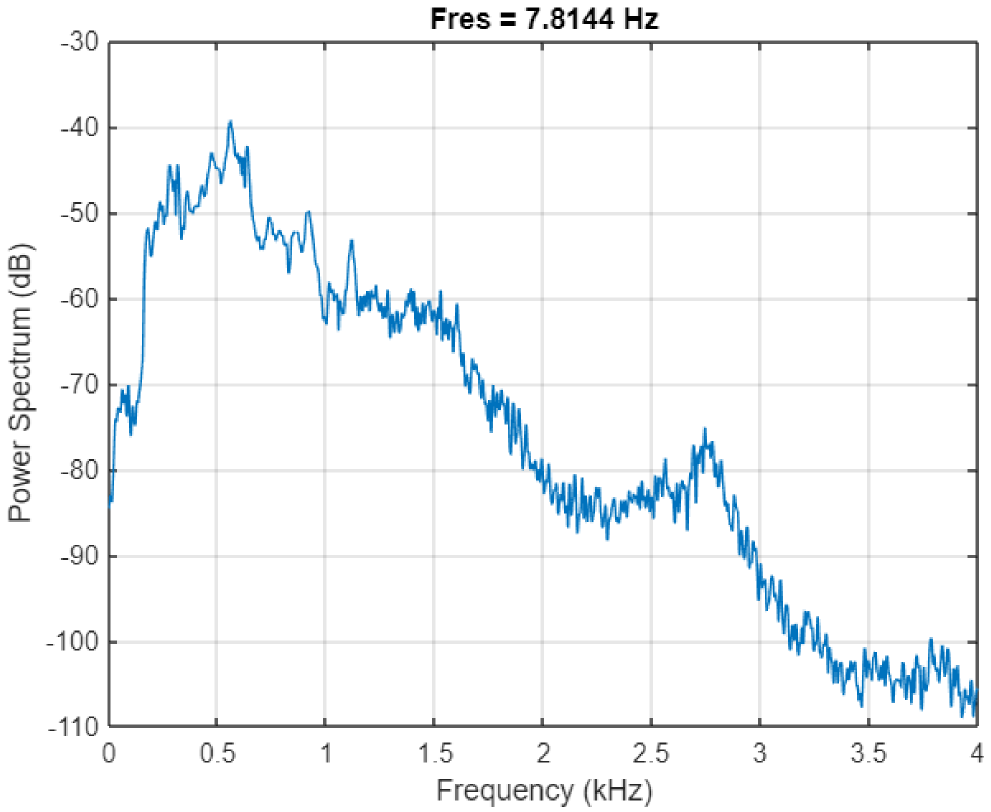


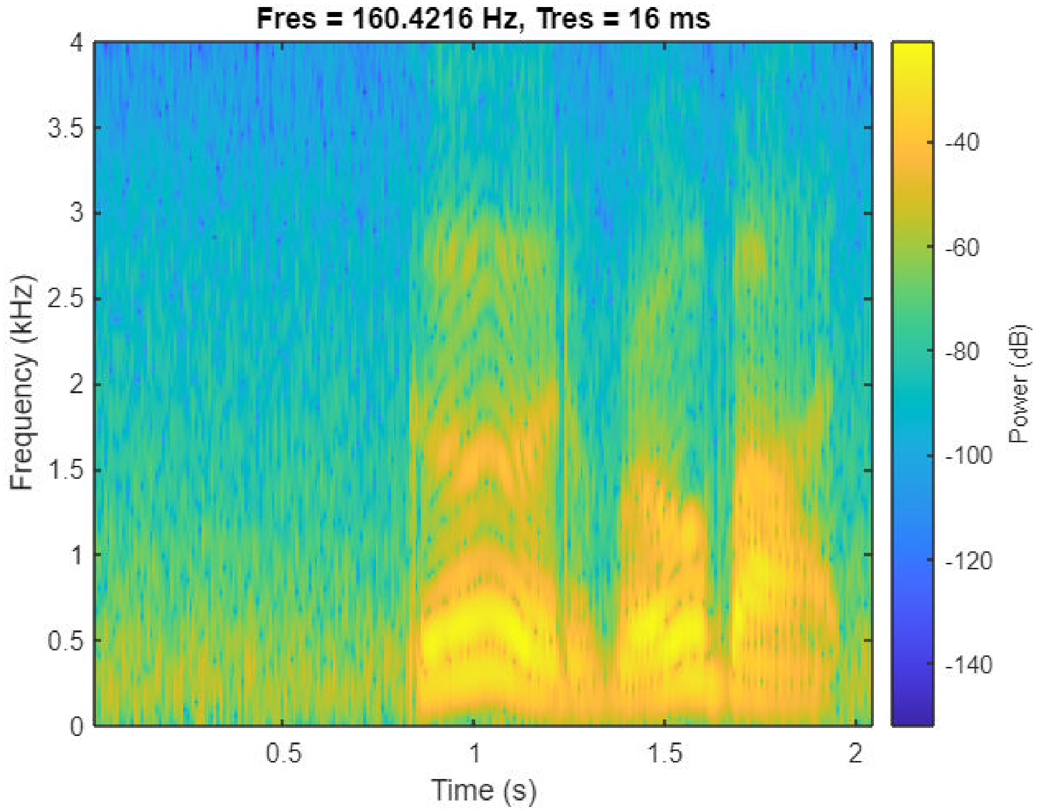
Prohibition


Fres = 7.8144 Hz
Power Spectrum (dB)
-40
-50
-60
-70
-80
-90
-100
-110
0
0.5
1
1.5
2
2.5
3
3.5
4
Frequency (kHz)


Fres = 160.4216 Hz, Tres = 16 ms
Frequency (kHz)
4
3.5
3
2.5
2
1.5
1
0.5
0
0.5
1
1.5
2
Time (s)
-40
-60
-80
-100
-120
-140
Power (dB)

Approval


Fres = 7.8144 Hz
-30
-40
-50
-60
-70
-80
-90
-100
-110
Power Spectrum (dB)
0
0.5
1
1.5
2
2.5
3
3.5
4
Frequency (kHz)


Fres = 160.4216 Hz, Tres = 16 ms
4
3.5
3
2.5
2
1.5
1
0.5
0
Frequency (kHz)
0.5
1
1.5
2
Time (s)
-40
-60
-80
-100
-120
-140
Power (dB)

2.3 Visualize relevant features of the audio signals

The following LiveScript Task allows the user to interactively choose audio features from a given signal. This automatic extraction will be used later to create the training and test features. Here the Live Task is just reported to show which audio features can be extracted and which parameteres ca be tuned.

A brief explanation of some relevant audio features can be found in the following source:

Devopedia. 2021. "Audio Feature Extraction." Version 8, May 23. Accessed 2021-09-09. https://devopedia.org/audio-feature-extraction

```
Output column mapping

        melSpectrum: 1:32
               mfcc: 33:45
   spectralCentroid: 46
      spectralCrest: 47
   spectralFlatness: 48
       spectralFlux: 49
   spectralSkewness: 50
      spectralSlope: 51
```

3. Signal Preprocessing

Signals are preprocessed, accomplishing two different goals:

- removing the voice-free part at the beginning and at the end of the signal, to keep only the relevant portion of the audio recording
- smoothing the signal using a moving-average filter, to remove part of the measurement noise

In order to remove the voice-free portion of the signals, the spectral energy of the signal has been computed only samples corresponding to values above a given threshold have been kept.

Some samples for which the algorithm does not detect significant spectral energy values are removed as well, in order to obtain a cleaner dataset.

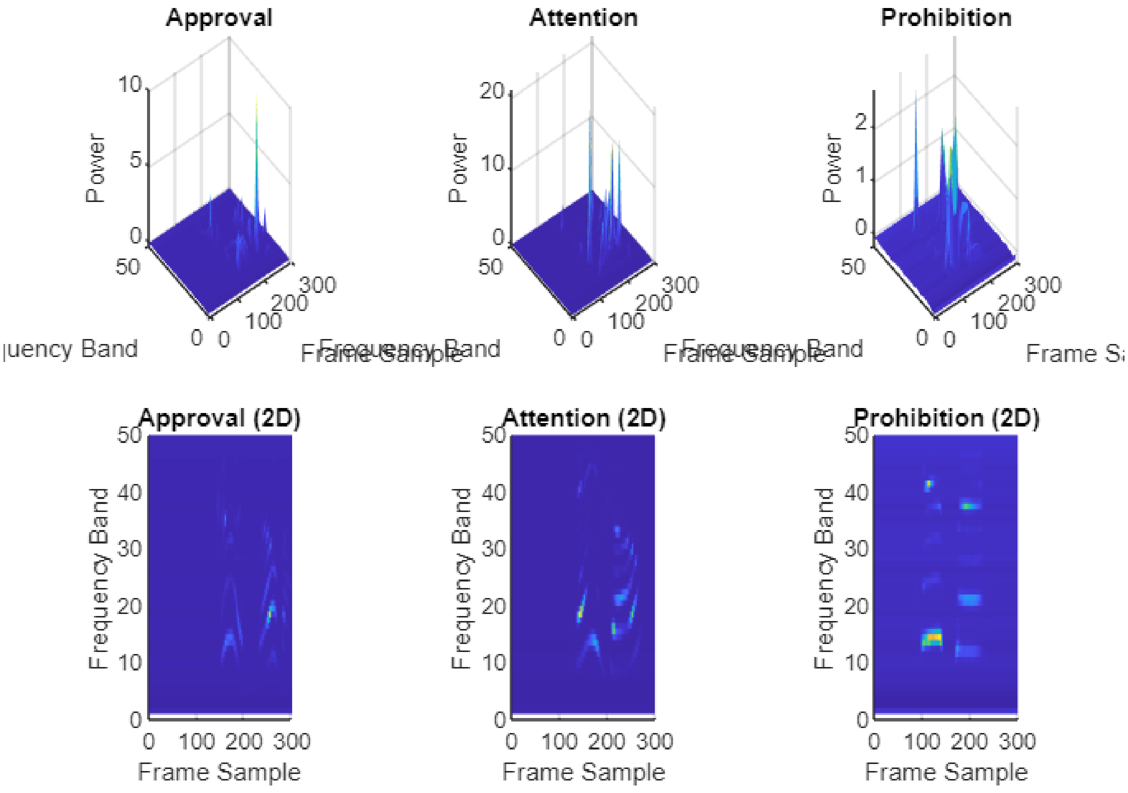
```
Removed sample #111
Removed sample #128
Removed sample #152
Removed sample #156
Removed sample #158
Removed sample #33
Removed sample #50
Removed sample #75
Removed sample #80
Removed sample #85
Removed sample #86
Removed sample #103
Removed sample #121
Removed sample #121
Removed sample #132
Removed sample #135
Removed sample #135
Removed sample #138
Removed sample #149
Removed sample #154
Removed sample #161
Removed sample #164
Removed sample #166
Removed sample #167
Removed sample #169
Removed sample #169
Removed sample #169
Removed sample #169
Removed sample #171
Removed sample #171
Removed sample #171
Removed sample #171
Removed sample #172
Removed sample #172
Removed sample #172
Removed sample #180
Removed sample #180
Removed sample #183
Removed sample #183
Removed sample #185
Removed sample #187
Removed sample #188
Removed sample #188
Removed sample #189
Removed sample #190
Removed sample #192
Removed sample #192
Removed sample #192
Removed sample #194
Removed sample #199
Removed sample #201
Removed sample #203
Removed sample #203
Removed sample #205
Removed sample #206
Removed sample #211
Removed sample #216
Removed sample #216
Removed sample #219
Removed sample #221
Removed sample #221
Removed sample #221
Removed sample #223
Removed sample #227
Removed sample #229
Removed sample #232
Removed sample #257
Removed sample #258
Removed sample #258
Removed sample #258
Removed sample #280
Removed sample #280
Removed sample #280
Removed sample #290
Removed sample #290
Removed sample #290
Removed sample #295
Removed sample #295
Removed sample #303
Removed sample #303
Removed sample #303
Removed sample #303
Removed sample #303
Removed sample #303
Removed sample #304
Removed sample #306
Removed sample #306
Removed sample #306
Removed sample #307
Removed sample #307
Removed sample #307
Removed sample #310
Removed sample #311
Removed sample #311
Removed sample #311
Removed sample #311
Removed sample #313
Removed sample #314
Removed sample #314
Removed sample #314
Removed sample #315
Removed sample #318
Removed sample #319
Removed sample #319
Removed sample #321
Removed sample #322
Removed sample #322
Removed sample #323
Removed sample #323
Removed sample #323
Removed sample #342
Removed sample #354
Removed sample #357
Removed sample #364
Removed sample #366
Removed sample #366
Removed sample #366
Removed sample #366
Removed sample #367
Removed sample #368
Removed sample #369
Removed sample #370
Removed sample #372
Removed sample #374
Removed sample #374
Removed sample #378
Removed sample #378
Removed sample #379
Removed sample #382
Removed sample #383
Removed sample #383
```

4. Feature Extraction

Features are extracted from preprocessed signals following two different approaches:

- Extraction of multiple spectral features, to be fed separately to a Neural Network for classification
- Extraction of the spectrograms of the audio recordings and their resizing to equal dimensions, so that they can be seen as images to be fed to the *imageInputLayer* of the Neural Network

Both approaches have been tested, but in the following sections only the second one is developed.



5. Training and Test Datasets, Feature Normalization and PCA

Datasets are split into training and test sub-sets, according to different criteria:

- Intra-corpus approach
- Cross-corpus approach
- Pooling approach

Features are normalized with respect to training data and, just in the case of multiple spectral features and not of plain spectrograms, a PCA is carried out to understand which predictors contribute the most in explaining the total variance in the dataset.

Since here we are dealing with the spectrogram approach, no PCA is executed.

5.1 Intra-corpus Approach

5.1.1 Train on Kismet, Test on Kismet

5.1.2 Train on BabyYears, Test on BabyYears

5.2 Cross-corpus Approach

5.2.1 Train on Kismet, Test on BabyYears

5.2.2 Train on BabyYears, Test on Kismet

5.3 Pooling Approach

5.3.1 Train on Kismet and BabyYears, Test on Kismet

5.3.2 Train on Kismet and BabyYears, Test on BabyYears

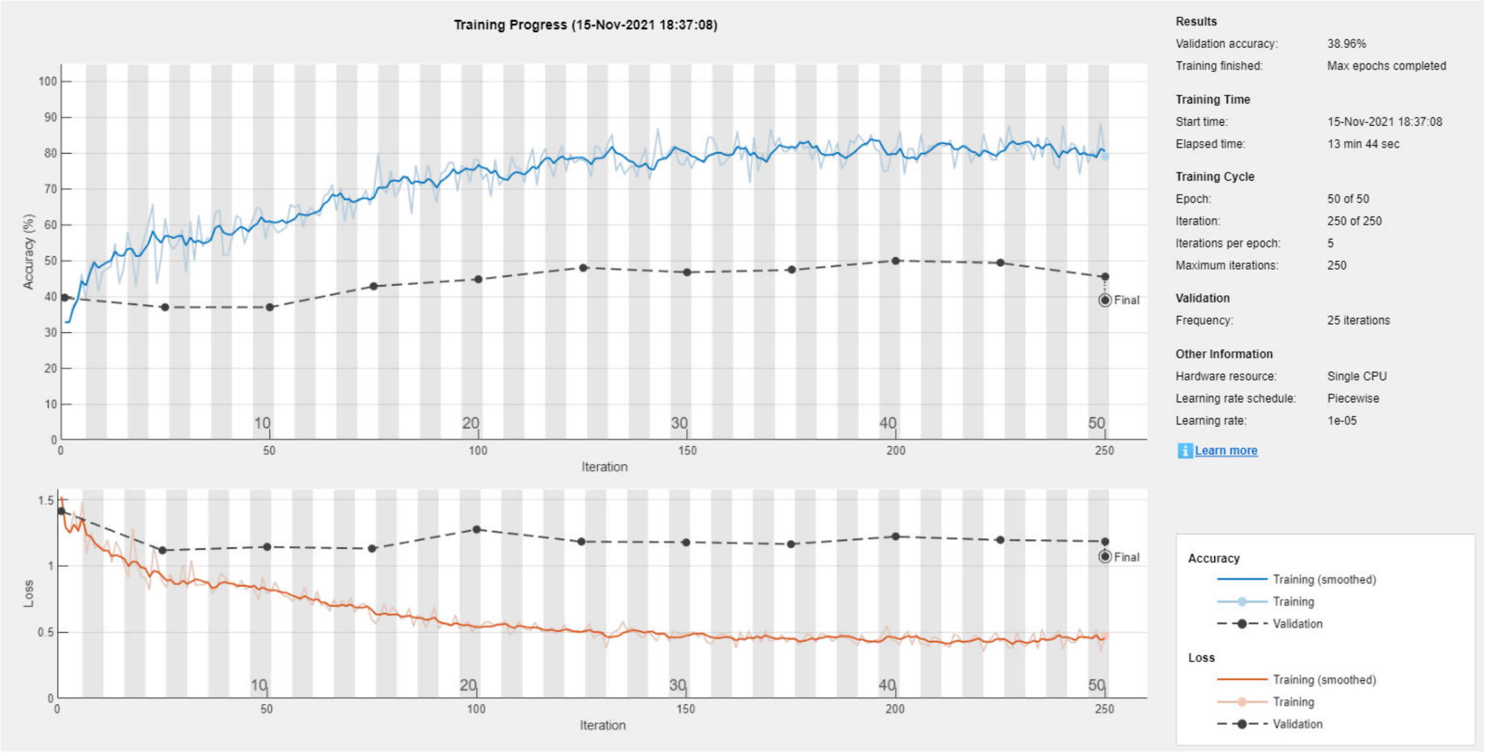
5.4 Normalization

5.5 Principal Component Analysis

Not relevant, since we are extracting spectrograms from the audio signals and we are classifying them as images.

6. Training of the Classifier

The Neural Network is trained and is saved, so that a different network for each different approach can be later tested on new data and evaluated.



Results

Validation accuracy: 38.96%

Training finished: Max epochs completed

Training Time

Start time: 15-Nov-2021 18:37:08

Elapsed time: 13 min 44 sec

Training Cycle

Epoch: 50 of 50

Iteration: 250 of 250

Iterations per epoch: 5

Maximum iterations: 250

Validation

Frequency: 25 iterations

Other Information

Hardware resource: Single CPU

Learning rate schedule: Piecewise

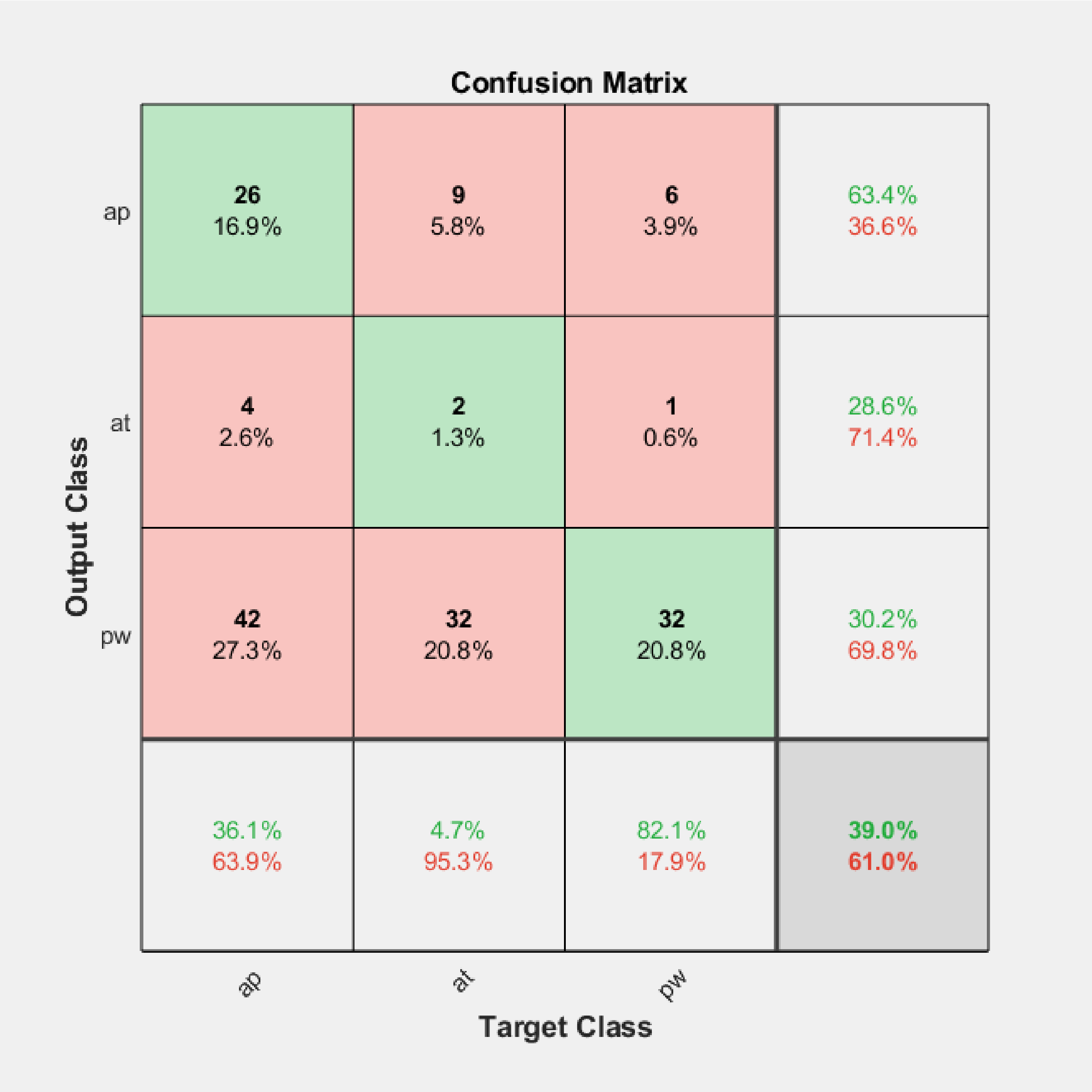
Learning rate: 1e-05

7. Validation of Classifier Accuracy (Bootstrapping)

7.1 Compute Validation Accuracy and Plot Confusion Matrix

Validation accuracy and the corresponding confusion matrix are shown below. They are referred to a single run performed over the entire test dataset that has been provided to the network.

```
accuracy = 38.9610
```



	ap	at	pw	
ap	26 16.9%	9 5.8%	6 3.9%	63.4% 36.6%
at	4 2.6%	2 1.3%	1 0.6%	28.6% 71.4%
pw	42 27.3%	32 20.8%	32 20.8%	30.2% 69.8%
	36.1% 63.9%	4.7% 95.3%	82.1% 17.9%	**39.0%** **61.0%**

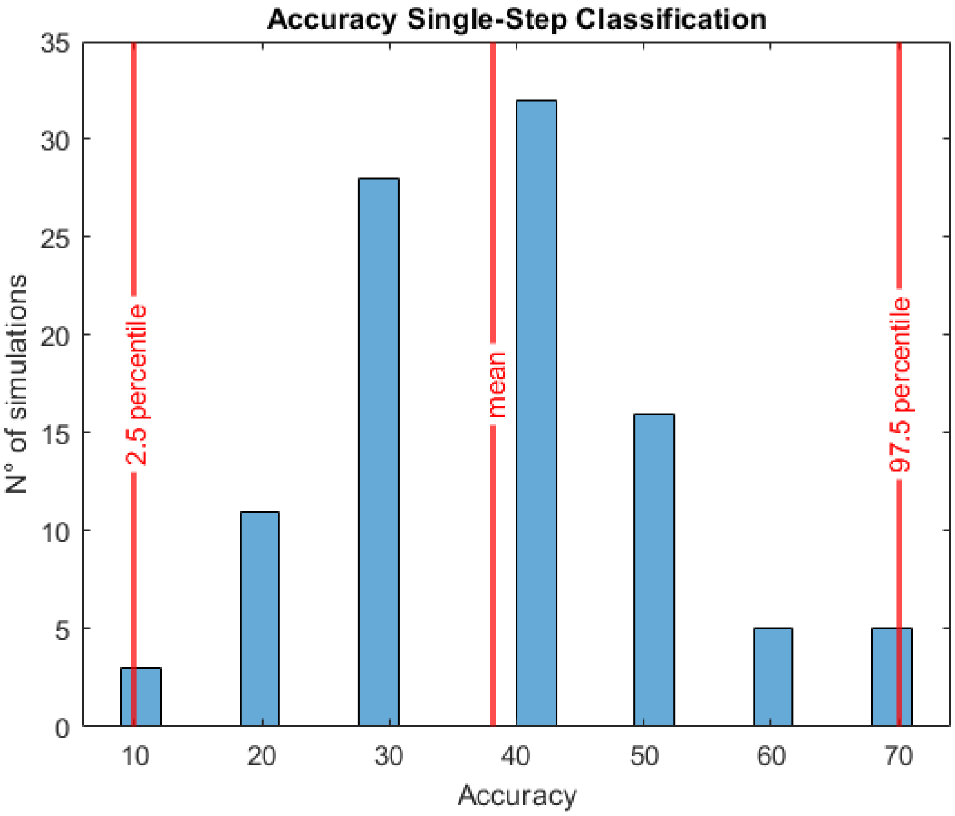
7.2 Evaluate Network Size, Prediction Time and Accuracy using Bootstrapping

The performance of the network is evaluated in terms of:

- size
- average prediction time
- average prediction accuracy and confidence interval (95%)

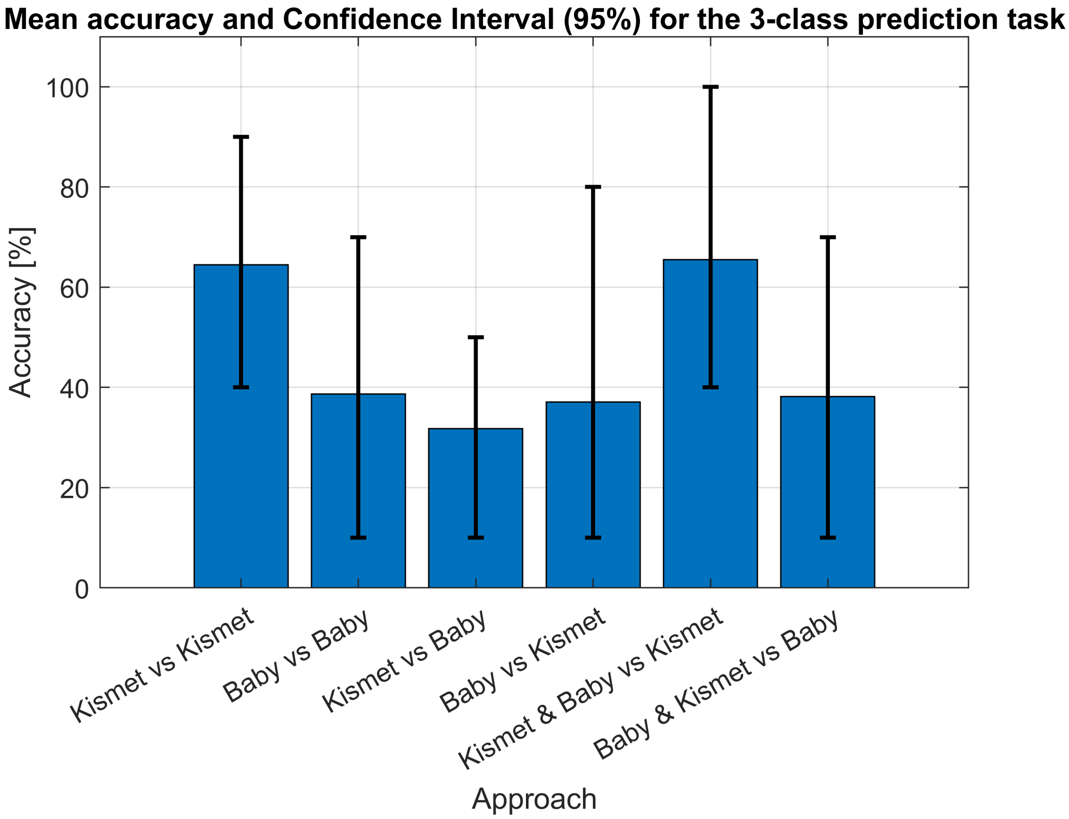
100 runs are executed over randomly drawn subsets of the test samples, in order to average out results and obtain an accuracy distribution across multiple runs. The mean accuracy is then taken as the final metric for evaluating the network.

```
Network size: 293.5986 kB
Single-sample prediction time on CPU: 40.2931 ms
Mean accuracy: 38.2 %
Confidence Interval: [10, 70]
```



8. Comparison among different Approaches

The six different approaches are compared: mean accuracies and confidence intervals over multiple runs are shown. Under the same values of the parameters for network building and training and the same window length and overlap for the spectrogram extraction, it can be seen that results are generally very poor (mean accuracies ranging from 30% to 65%). It can be seen that the CI are very wide, meaning that accuracy results can vary a lot depending on the different test sub-datasets the net is provided with: the model generalizes badly, hence overfitting is an issue to deal with.



Mean accuracy and Confidence Interval (95%) for the 3-class prediction task

Conclusions

This short work provides a framework for classifying audio signals through image recognition of their spectrogram. However, accuracy and generalization results are poor, hence improvements have to be made. The possible areas on which to intervene are:

- proper dataset selection for the training phase,
- better tuning of the network, both in terms of its structure and of the training options,
- more accurate feature extraction process, resulting in more meaningful features that allow for a better separation of the classes. In particular, a better tuning of the window length, overlap percent, frequency/time resolution can be carried out,
- different ways to preprocess signals, in order to better filter out irrelevant information and keep only the most meaningful portion of the recording.