

Comparison of Time-Frequency Representations for Neural-Network based Keyword Spotting

Lukas Till Schawerda

In cooperation with the Acoustics Research Institute at the Austrian Academy of Sciences

THE CONTEXT

WHAT is keyword spotting?

Literally any other word



"Alexa"



Recognizing a specific word or command in a stream of speech audio data

Common applications: Smart home devices (Alexa, Siri, Hey Google, etc.)

WHY is it interesting?

Diverse user base
languages, accents, voice ranges
Diverse environments
room acoustics, background noise

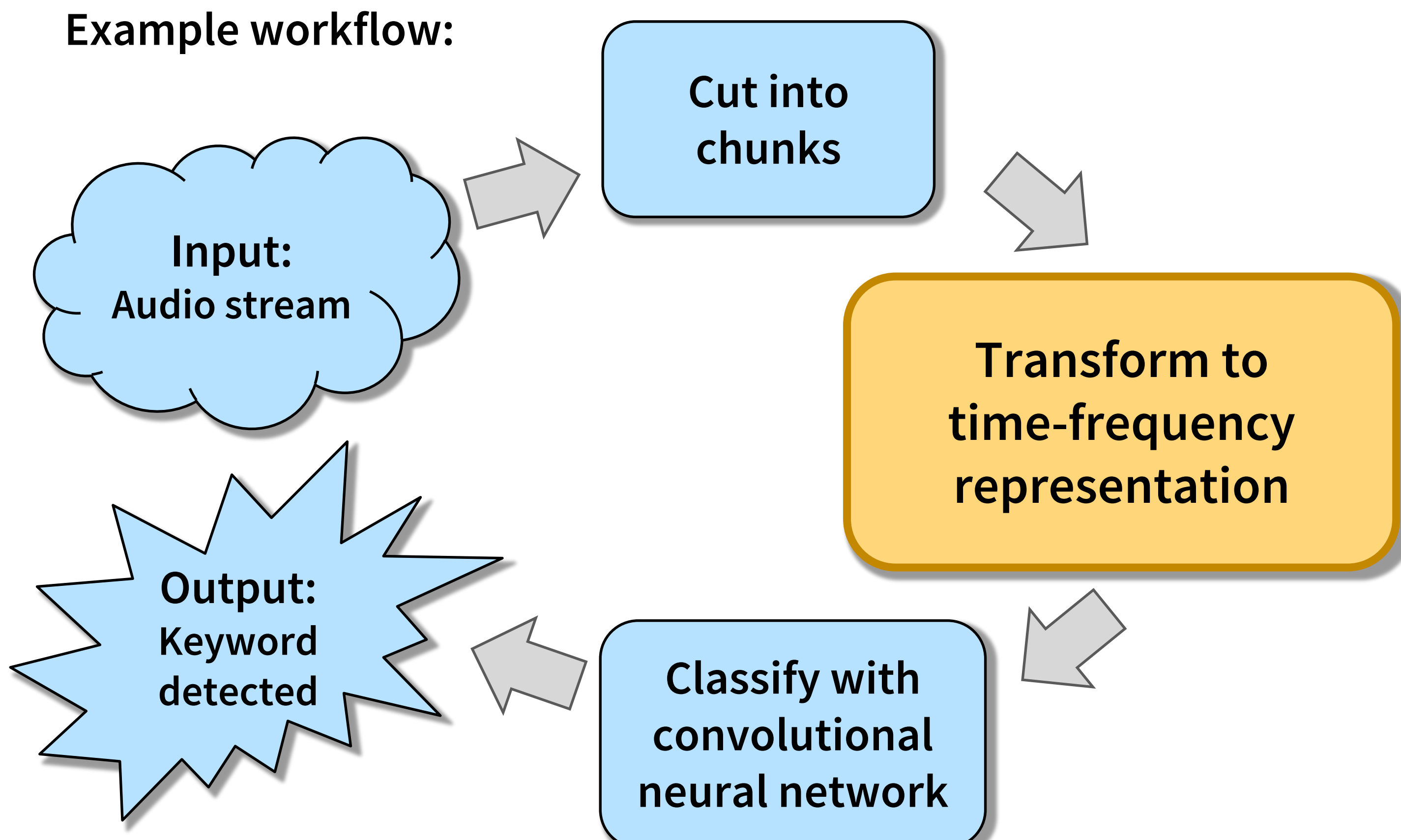
Runs on mobile devices

- Limited computational power
- Limited storage capacity
- Always on in the background

Models need to be strong and cheap → exciting design challenge!

HOW can it be addressed?

Example workflow:



THE PROJECT

Goals

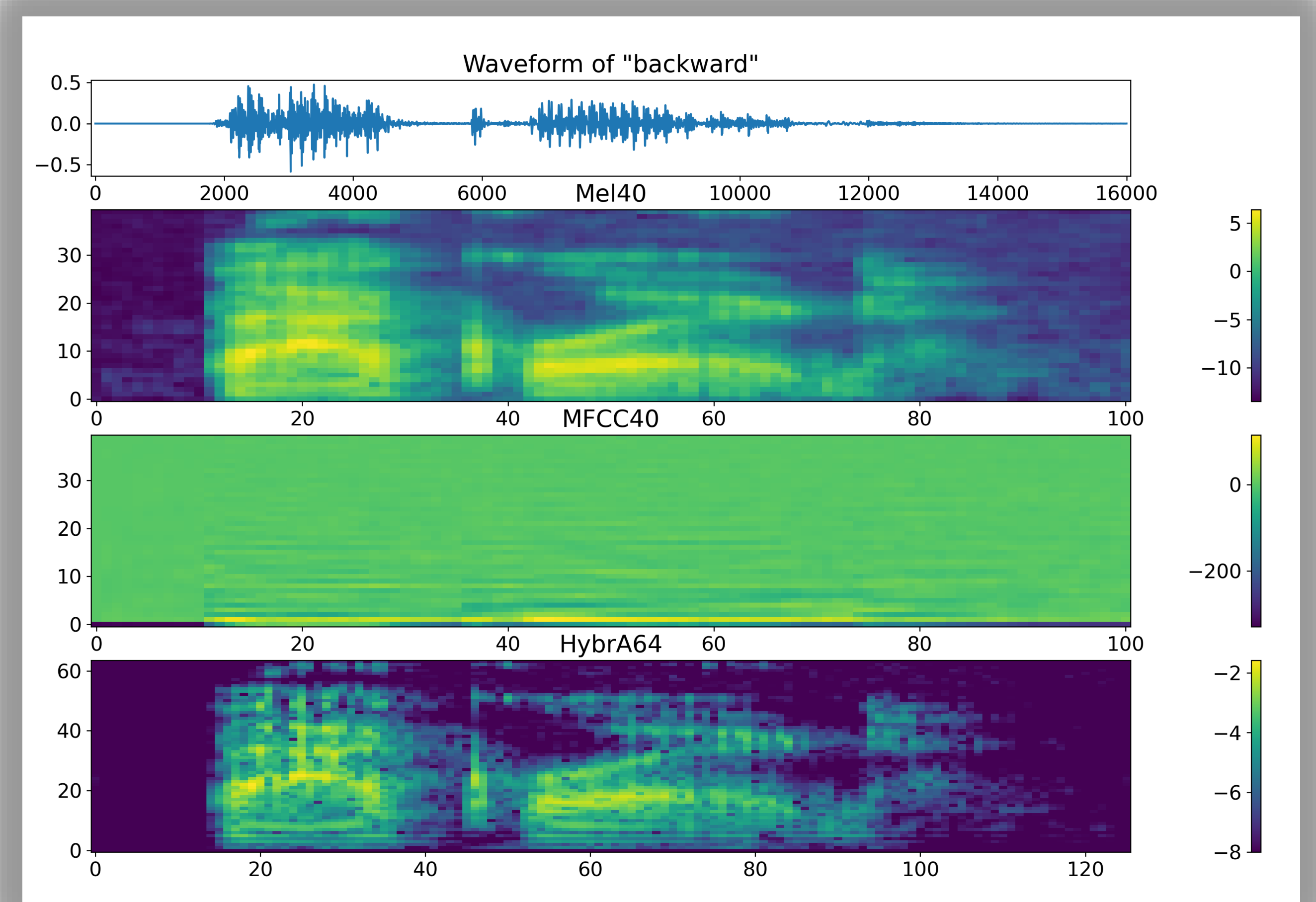
Test how different time-frequency representations (TFRs) affect neural network model performance in keyword spotting tasks

TFRs display the frequency composition of the original signal over time → Two dimensions, time + frequency

Example: Spectrogram

- Divide audio into windowed, overlapping subsections
- Perform Fourier transform

Experiments



- Three TFRs as inputs: Mel Spectrogram, Mel Frequency Cepstral Coefficients (MFCCs), and Hybrid Filterbanks (HybrAs)
- Model: BCResNet (three sizes ranging from 9.2k to 54.2k parameters)
- Dataset: Google SpeechCommands (12 classes)

Results

| Model | TFR | Train | Test |
|-------------------|-----------------|--------------|--------------|
| BCResNet-1 | Mel40 | 0.971 | 0.885 |
| BCResNet-1 | MFCC40 | 0.968 | 0.881 |
| BCResNet-1 | HybrA64 | 0.970 | 0.887 |
| BCResNet-1 | HybrA256 | 0.981 | 0.891 |
| BCResNet-2 | Mel40 | 0.950 | 0.895 |
| BCResNet-2 | MFCC40 | 0.944 | 0.894 |
| BCResNet-2 | HybrA64 | 0.945 | 0.894 |
| BCResNet-2 | HybrA256 | 0.957 | 0.891 |
| BCResNet-3 | Mel40 | 0.952 | 0.899 |
| BCResNet-3 | MFCC40 | 0.943 | 0.896 |
| BCResNet-3 | HybrA64 | 0.945 | 0.894 |
| BCResNet-3 | HybrA256 | 0.953 | 0.890 |

Test accuracy close to 0.9 for a twelve-class classification problem!

However, all TFRs appear to perform similarly well

Conclusion: The TFR chosen as input to the neural network seems to have only marginal impact on model performance