APAC-adabra

a deep learning model echoing the magic behind the human auditory cortex for fast and accurate speech recogntion

Nicola Cassetta
nicola.cassetta@studenti.unipd.it

06 September 2023

Abstract

Deep neural networks (DNNs) have gained significant attention and success across diverse domains. DNNs have outperformed traditional signal processing strategies via convolutional neural networks (CNNs) and recurrent neural networks (RNNs) in conjunction with various spectrogram representations. We propose a novel DNN architecture for audio classification, inspired from the intricate structure of the human auditory cortex, named "APAC" (Auxiliary Pathway Auditory Cortex). By emulating the underlying principles of the brain's auditory processing, our approach aims to enhance the performance and efficiency of audio classification tasks. This study highlights the potential for synergistic insights between neuroscience and machine learning in the pursuit of more effective and biologically-inspired AI systems. We analyze the performance of our model and make a comparison between other classical nets using different spectrogram representation.

1 Introduction

Deep neural networks (DNNs) are a class of machine learning models, composed of interconnected nodes organized in layers. In the last 20 years they have gained significant attention and success in various fields of artificial intelligence with breakthroughs in different areas, from image and speech recognition [1] [2] to natural language processing [3] and many others, mainly due to their ability to learn complex patterns and representations from large amounts of data. These models have demonstrated state-of-the-art performance and have become the foundation of many cutting-edge applications in academia, industry, and various real-world use cases, thanks also to the availability of both large datasets and computational resources. In the context of signal processing, the last decade has shown how deep learning models can outperform traditional signal processing strategies (deep CNNs and RNNs over Gaussian mixture models and Hidden Markov Models), with also a shift of the focus from traditional handcrafted features, to various spectrogram representations. This trend has been driven by the remarkable ability of DNNs to automatically learn intricate patterns and representations directly from raw audio data, and led to improved performance and more accurate classification results in various audio domains, including speech recognition [2], music genre classification [4], and environmental sound recognition [5].
For what concerns the process of designing the architecture of a deep leaning model, there are no strict rules, but rather rule of thumbs that helps researchers and practitioners in this complex task. One way to overcome this limitation is to base the construction of a new net taking into account the structure of the most complex computer: the brain. In this work we present the result of a novel deep neural network for audio classification, inspired from the human's auditory cortex.

2 Related works

The trend of designing deep neural networks by taking inspiration from various aspects of brain structure and functioning has significantly impacted the field of deep learning since the 90s. This interdisciplinary approach, combining insights from neuroscience and machine learning, has yielded improved performance and capabilities in a wide range of tasks. It is sufficient to take into account some of the most powerful and popular techniques used nowadays like the attention mechanisms: inspired by the human visual system, attention mechanisms allow neural networks to selectively focus on relevant features or regions in the input, improving their ability to process and understand complex data [6] [7] [8]. In the last decade advancements in neuroscience and cognitive psychology have provided insights into how the brain processes sensory information and performs computations, and thus, researchers have integrated these findings into neural network designs.

3 Mel-spectrogram

Traditional audio classification approaches required expert knowledge and manual engineering, involving the extraction of specific characteristics believed to be discriminative for the tasks. However, the effectiveness of handcrafted features heavily relied on the assumptions and biases introduced during the feature engineering process. Spectrograms, on the other hand, provide a visual representation of the audio signal's time-frequency content, allowing deep learning models to capture rich and complex patterns.

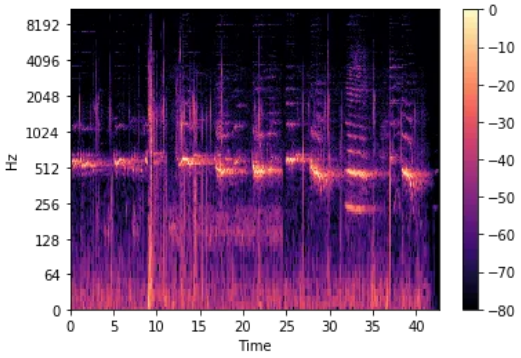

Figure 1: Spectrogram of the haunting song of humpback whales

Among them, one of the most important is the Mel-Spectrogram: a type of spectrogram that uses the Mel scale to represent frequencies. More precisely: as for standard spectrogram we map the input signal from the time domain to the frequency domain via the short-time FFT (Fast Fourier Transform) and convert the frequency-axis to log scale, and the amplitude-axis to Decibels, then we add another step that is the convertion of the frequencies in Mel-scale. This version quickly gained popularity due to its ability of approximate the human auditory perception (the use of the Mel scale helps mimic the fact that humans do not perceive frequencies on a linear scale but rather in a logarithmic scale, resulting in the ability of detecting differences in lower frequencies more accurately than differences in higher ones). By converting the raw audio signal into a Mel-spectrogram, deep models could focus on capturing the essential frequency components that are more perceptually relevant.

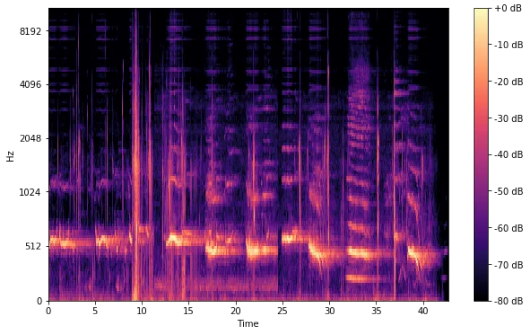

Figure 2: Mel-Spectrogram of the same audio of Fig1

Moreover the transition from handcrafted features to spectrogram-based representations facilitated a data-driven approach to audio classification, additionally, the adoption of spectrogram-based representations aligns with the trend towards end-to-end learning in deep learning models: by feeding the raw audio signal or its spectrogram directly into the model, the entire classification pipeline can be trained jointly, optimizing the model's parameters for the specific task at hand.

3.1 Mel-spectrogram via 1D-CNN

In 2020 Kin Wai Cheuck and its team released "nnAudio", a neural network based audio processing toolbox library based on PyTorch that leverages 1D CNN to generate spectrograms on-the-fly, allowing also for the tuning of the parameters involved in the spectrogram and the filterbank generation (for context, Kapre has a similar approach in which they also use 1D CNN to produce spectrograms, but the work is based on Keras [9]). The detailed description of how this process works lies outside the goal of this paper, thus we highly suggest to read the section III-A of [10] or to explore the public repo in order to fully understand how they compute the STFT (Short Time Fourier Tranform) via a pair of convolutional kernel.

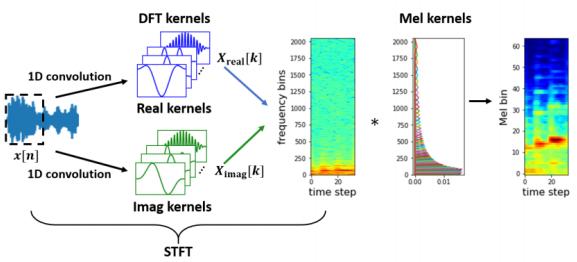

Figure 3: nnAudio's neural network-based implementation for Mel spectrograms

In this project we will compare the results obtained via this method to the results obtained via the classical spectrogram generation (specifically, in the latter case, we will use the torchaudio library [11]).

4 MFCC

The MFCC is an extremely popular feature representation method that transforms a raw audio waveform into a compact representation, facilitating efficient analysis and classification of audio signals. It draws upon both the mel scale, which, as mentioned in the previous section, approximates the human auditory system's frequency perception, and the cepstral analysis, which separates the spectral characteristics of the signal from its temporal characteristics. The computation of this representation is quite complex, and the process can be summarized into 5 different steps:

- Framing: the audio signal is divided into short overlapping frames
- Power spectrum: compute the power spectrum of each single frame
- Mel filterbank: bank of triangular filters on the Mel scale is applied to the power spectrum
- Log compression: needed to approximate the log perception of loudness by the human auditory system
- Discrete Cosine Transform (DCT): this step aims to both decorrelate filterbank energies (since filterbanks are quite overlapping) and also to discard some informations

The MFCC representation still plays nowadays an important role in preparing audio data for deep learning-based audio recognition tasks thanks to its ability to extract essential characteristics, and thanks to its dimensionality reduction properties [12].

5 Human auditory cortex

The human auditory cortex is a specialized region within the brain (specifically located in the temporal lobe) that plays a fundamental role in processing auditory information and facilitating our perception of sound, ranging from sound localization to music processing. It is hierarchically organized and comprises multiple interconnected areas that process different aspects of auditory stimuli. This organization allows for the extraction of diverse features from sound signals. The AC (Auditory Cortex) presents two main regions:

- Primary Auditory Cortex (A1): Located within the temporal lobe, it receives raw auditory signals from the thalamus and is responsible for basic sound analysis.
- Secondary auditory cortex (A2): higher-order auditory areas (like the Belt and Parabelt) that perform more complex analysis of the auditory information.

within the auditory cortex, there is a layered structure of neurons. The layers are numbered from 1 to 6, with layer 1 being the outermost layer and layer 6 being the deepest one. Despite this division, the structure is not feed-forward, but the information are shared back and forth among the different layers. Each layer has a specific structure and goal, but all of them have in common the presence of pyramidal and stellate cells: two types of excitatory neuron. The presence of six cell layers in the auditory cortex is common to all mammals, but species differences take the form of the commonality of each cells within each layer. In humans, pyramidal cells correspond to

85% of A1, the remaining 15% are stellate cells [13] [14] .
In this work we take inspiration from the structure of AC rather than randomly search a good architecture or take inspiration from older structures (like the well-known VVGish, ResNet, WaveNet, and so on), to build a deep learning model tailored for the audio classification task.

6 APAC architecture

In this section we will first list and describe all the modules that are implemented in our net, then we will describe how they are organized and how they are related to the AC.

6.1 Modules

6.1.1 CBAM

Initially proposed by [6], the Convolutional Block Attention Module (CBAM) is nowadays a popular attention mechanism: it consists of two modules, that are the Channel Attention Module (CAM) and the Spatial Attention Module (SAM). The CAM aims to capture channel-wise infos (the "what"), the SAM, on the other hand, focuses on capturing spatial dependencies (the "where") by learning to attend to relevant spatial locations. By integrating both channel-wise and spatial-wise attentions, the CBAM block allows CNNs to dynamically adapt their feature representations to improve both channel and spatial information utilization.

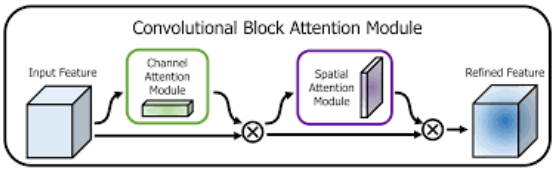

Figure 4: CBAM block

6.1.2 SE block

Another very popular attention module is the SE (Squeeze-and-Excitation) block, proposed by [7]. This module consists of two main operations: the squeezing and the exciting. In the first one, global information is captured by applying global average pooling to the input feature maps, reducing the spatial dimensions to a single value per channel. This step enables the network to summarize the channel-wise information. In the exciting phase, the channel-wise information is recalibrated using a set of fully connected layers. These layers generate channel-wise attention weights. The adjusted feature maps are obtained by multiplying the original feature maps with the channel-wise attention weights.

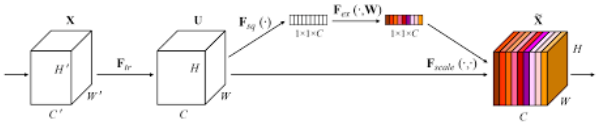

Figure 5: SE block

6.1.3 Separable Convolution

The MobileNet [15] pointed out a clever decomposition of a standard convolution operation into two efficient operations: depthwise convolution and pointwise convolution. In the first operation, each input channel is convolved with a separate filter, generating a set of intermediate feature maps. This operation applies a spatial filter to each input channel independently. The second part applies a 1x1 convolution, where the intermediate feature maps from the depthwise convolution are linearly combined to create the final output feature maps. With this simple modification the overall computational complexity of the convolutional layer is greatly reduced.

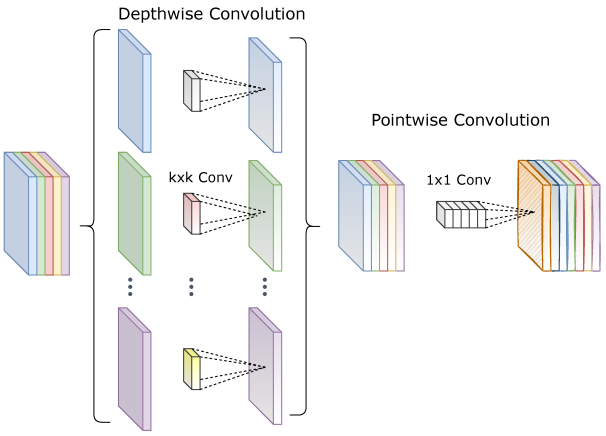

Figure 6: Separable convolution

6.1.4 Inverted Residual Block

Firstly introduced in the MobileNetV2 [16] architecture with the aim of enhancing the efficiency of CNNs without sacrificing accuracy, the Inverted residual block is nowadays a very popular module capable of optimize in a plug-and-play approach almost any CNN. By breaking up the structure of this module we have different steps:

- Expansion: increase the number of channels with a 1x1 conv layer. The expanded channels allow the model to capture more diverse and complex features.
- Depth-wise Separable Convolution: already discussed in this work, this technique helps to capture spatial correlations and extract meaningful features in a more efficient way.
- Linear Bottleneck: again a 1x1 convolution that compresses the number of channels back to the original size, preparing the feature map for the skip connection (it also effectively reduce the computational complexity and memory footprint)
- Skip Connection: Similar to traditional residual blocks, the Inverted Residual Block incorporates a skip connection: this allows the original input to be added to the output of the linear bottleneck. The skip connection facilitates gradient flow during training and helps to mitigate the risk of information loss.

Following the work of [16] we also added a BatchNormalization layer after each convolution and used the ReLU6 as activation function within this module (the ReLU6 is simply a ReLU with a max value allowed of 6).

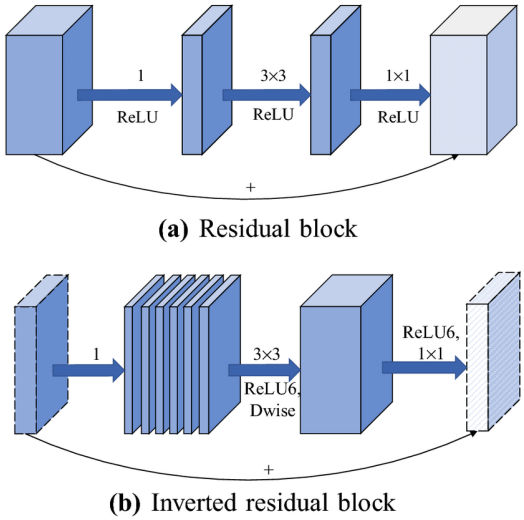

Figure 7: Comparison between a classical residual block (a) and the Inverted residual block (b)

6.2 Architecture

As we write this paper, the human knowledge still lack in fully understanding how the brain works. As a consequence, also the knowledge on how the informations flows in the AC and how they are managed is not fully complete. Despite that, there are many works that try to summarize or describe the structure of this complex system, among which we must cite [14]

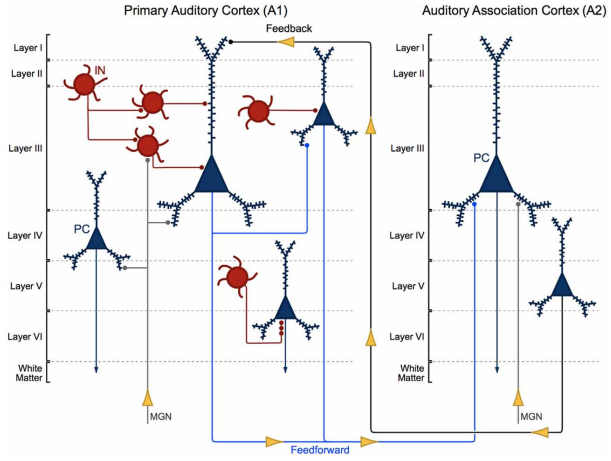

Figure 8: Rough auditory pathway scheme taken from the work of [14]

The lack of these informations does not allow for a perfect replica of the structure of the AC, and this is the main reason of why we use the term "inspired" for our net. In this work we took some concepts that are fundamental for the functioning of the AC and we tried to map them in the field of deep nets, specifically:

- The presence of two macro-areas that jointly works on the same input but with different strategies (A1 and A2) has been translated into two main path. The A1 is mapped as our primary path, and uses the mel-spectrogram as input, while the A2 is the auxiliary pathway that leverages the MFCC to help the correct classification of the audio.
- Inside the A1 there is a plethora of neurons that behave in different way in order to extract different levels of informations from the same input. This concept has been translated in the presence of two different CNN-branches that leverages two different kernel sizes (3x3 and 7x7) that helps to works at different granularity on the same melspectrogram. The resulting features map are than merged.
- The presence of mainly two types of excitatory neurons (Stellate and Pyramidal) has been translated with the presence of two different attention modules (CBAM and SE).
- Despite the fact that a real-time inference was not the goal of this work, this is still a feature of the AC. We used the separable convolutions and the inverted residual blocks to speed up the net (gaining a reduction on the training/inference time of 2x and this also helped in drastically reduce the number of parameters).
- As mentioned in the previous points, there are different types of neurons inside the AC. Inhibitory neurons are also an important component since they allow for the suppression of non-relevant informations. Trying to produce or search for a stand-alone module with the same task was quite difficult and in the end we managed to map this concept with the presence of weights decay, batch normalization, layer normalization and dropout.

Additionally, by taking inspiration from the ConvNext [17], there are some other micro-architecture modification that we put in place in order to improve the performance of the net, among which we cite: the use of the GeLU instead of the ReLU on the features extraction phase, fewer activation functions and fewer batch - layer normalization.

For readability the detailed scheme of the architecture is in the section 11 "Additional material".

7 Dataset

7.1 Speech Commands

The Speech Command dataset [18] is a common and widely recognized collection of audio samples mainly used for speech recognition tasks. The dataset comprises recordings obtained from various speakers, ensuring diversity in terms of accents, tones, and pronunciations. An appealing characteristic of this dataset is that it comes already balanced and divided into "train", "validation" and "test". For this work we used the updated version of the dataset (named

"v2"), composed of 35 classes (instead of 10) ranging from classical audio like "one", "yes", "stop" to newest "learn", "house" and "visual". The train set is made up of 84843 samples, the validation has 9981 samples and the test set has 11005 samples. All the samples are already in Mono and with a sample rate of 16k.

7.2 Free Spoken Digit Dataset

A simple open audio/speech dataset consisting of recordings of spoken digits in wav files at 8kH, currently involves 6 speakers with English pronunciations and 3,000 recordings (50 of each digit per speaker) with 10 classes [19].

8 Experimental setup

In this section we will discuss all the models and the tests that have been performed.

Models			
VGG19	**ResNet50**	**MobileNetV2**	**APAC**

Table 1: List of all the models that we used

Starting from the models and recalling the Tab.1, we highlight the fact that all of them have been tested with and without pre-training (on ImageNet1k) on the same dataset. All the tests have been carried out with Adam as optimizer, learning rate equal to $1e^{-3}$ and a value for the weight decay equal to $1e^{-3}$ for 80 epochs for the models without pretraining, while in the other case we used the same optimizer but with a lower learning rate ($1e^{-4}$) and a lower number of epochs (30). We also point out that the results for the APAC net will be presented in five different scenarios:

- Non trainable Mel-spectrogram: classical approach, in this case we do not rely on the nnAudio library
- Trainable Mel filterbank
- Trainable STFT
- Trainable Mel + STFT
- Fine tune trainable Mel + STFT pretrained on the Digits dataset

We kept the same hyperparameters in all the scenarios: 2048 as n_fft, 128 as n_mels for the melspectrogram and 64 for the MFCC and 512 as hop_length.

9 Results

In this section we will show the result of all the models on the training/validation set, and in the section 9.1 we will present the results of the best models on the Test set. Starting with the classical models we have:

Model	P	Train acc	Val acc	Time
VGG19		0.96	0.78	230
ResNet50		0.94	0.72	175
MobNetv2		0.94	0.63	105
VGG19		0.97	0.88	
ResNet50		0.97	0.83	
MobNetv2		**0.95**	**0.87**	

Table 2: Train results for the standard nets. The field "P" stands for "Pretrained". Time is expressed in seconds

as we can see all the models are able to achieve quite good results, despite that we can appreciate the results of the pretrained MobileNetV2, since it is able to achieve the same results of the VGG19 (top-performance) within almost half of the time. Moreover it's also possible to notice an improvement on the generalization (val acc) on all the models thanks to the pretraining

APAC			
Mode	**Train acc**	**Val acc**	**Time**
Standard (std)	**0.91**	**0.92**	102
Mel	0.89	0.91	
STFT	0.88	0.88	
Mel + STFT	0.89	0.9	
Mel + STFT (Pretrain)	**0.9**	**0.91**	

Table 3: Results of the APAC net on the train and validation sets

Almost all the versions of the APAC net outperform classical models on the validation set, suggesting a better generalization and a better model overall. We will carry out the test phase with the two top-performing modes that are the Standard version and the Pretrained one.

9.1 Test set

At this point we are ready to test the models with the highest train/val accuracy value on the Test set. The models are the MobileNetV2 and the APAC_Net (via classical melspectrogram and via trainable melspectrogram)

Test set	
Model	**Accuracy**
APAC (std)	**91.4**
APAC (pretrain)	**90**
MobileNetV2	84.7

Table 4: Results of the best models on the test set

10 Conclusions

As mentioned at the beginning of this project, the practice of conceptualizing deep neural networks through diverse elements related to the structure and operational dynamics of the brain has exerted a substantial influence, and often resulted into better performing models.
By analyzing the results of the Tab. 2 and 3, we can clearly see that our net outperform almost all the classical net, even when pretrained on ImageNet1k: specifically we can notice that the highest value on the Val set for the standard models is the one of the MobileNetV2 (87%), while even the lowest value on the same set of data obtained by the APAC net is 88% (trainable STFT), while the highest is achieved with the standard computation of the melspectrogram and MFCC which leads to a 92% of accuracy.

If we then compare this two models on the test set, as showed in Sec. 9.1, we can see that the APAC outperform the MobileNet by 6-7% (depending on the mode). We also point out that the MobileNetV2 is a model notoriously fast, designed to work even on mobile devices, and the APAC net achieves a better result within the same amount of time!

By digging into the different modes of the APAC, we can notice that there are no substantial differences when switchin from the torchaudio library to the nnAudio one (even if it is possible to see that the results obtained via the classical melspectrogram computation are slightly higher if compared to the others). Despite that, one of the most interesting result is the one obtained by the "APAC (pretrained)": we can clearly see that even a simple pretraining step has led to an increment of a flat +1% on the accuracy. This suggest that - as for the classical neural networks - a more precise and carefully designed preprocessing step on the weights of the 1D-CNN could even result in a final model that outperform the classical spectrogram computation via torchaudio or librosa.

11 Additional material

In this section we will present some additional results and material. Starting from the structure and the legend for the APAC net, we have

11.1 APAC net

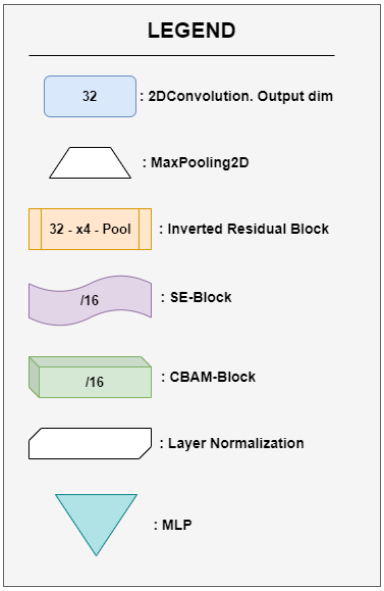

Figure 9: Legend needed for the APAC net scheme in Fig.13

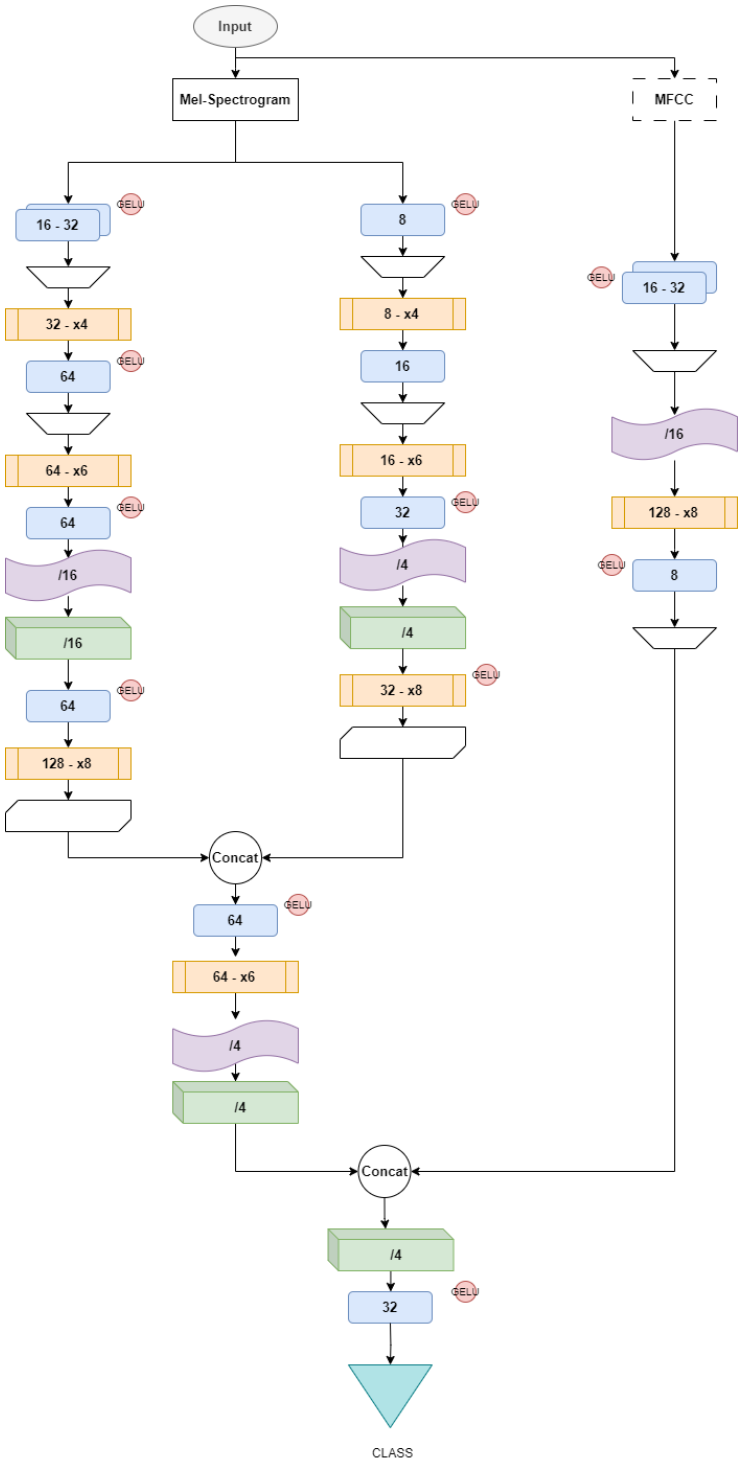

Figure 10: APAC net.

11.2 On the melspectrogram visualization

As a final analysis we show how the same audio is converted into a melspectrogram via the three different mode: the standard one (torchaudio), via the nnAudio in "cold start" (i.e. no training on the params) and with nnAudio after training and finetuning

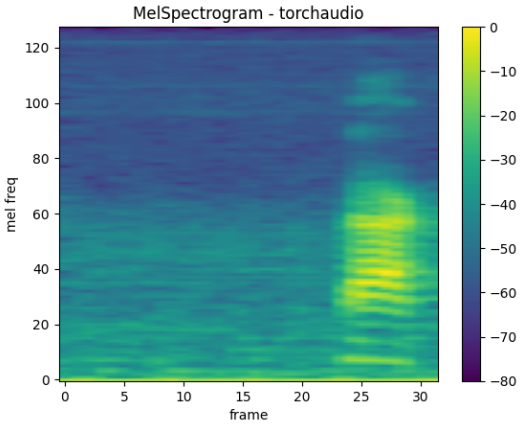

Figure 11: Torchaudio melspectrogram of the command "back"

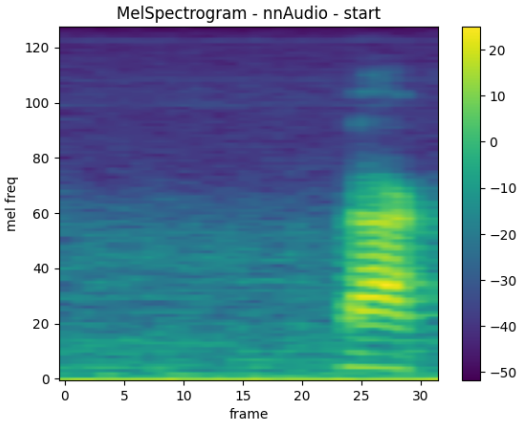

Figure 12: nnAudio melspectrogram cold start of the command "back"

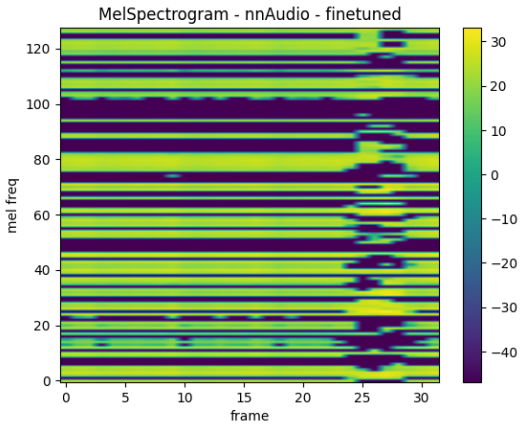

Figure 13: nnAudio melspectrogram after training and fine tuning of the command "back"

References

[1] S. Kriman, S. Beliaev, B. Ginsburg, J. Huang, O. Kuchaiev, V. Lavrukhin, R. Leary, J. Li, and Y. Zhang, "Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions," 2019.

[2] A. Mehrish, N. Majumder, R. Bhardwaj, R. Mihalcea, and S. Poria, "A review of deep learning techniques for speech processing," 2023.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[4] P. Ghosh, S. Mahapatra, S. Jana, and R. Jha, "A study on music genre classification using machine learning," *International Journal of Engineering Business and Social Science*, vol. 1, pp. 308–320, 04 2023.

[5] A. Bansal and N. K. Garg, "Environmental sound classification: A descriptive review of the literature," *Intelligent Systems with Applications*, vol. 16, p. 200115, 2022.

[6] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," 2018.

[7] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," 2019.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[9] K. Choi, D. Joo, and J. Kim, "Kapre: On-gpu audio preprocessing layers for a quick implementation of deep neural network models with keras," in *Machine Learning for Music Discovery Workshop at 34th International Conference on Machine Learning*, ICML, 2017.

[10] K. W. Cheuk, H. Anderson, K. Agres, and D. Herremans, "nnaudio: An on-the-fly gpu audio to spectrogram conversion toolbox using 1d convolution neural networks," 2020.

[11] Y.-Y. Yang, M. Hira, Z. Ni, A. Chourdia, A. Astafurov, C. Chen, C.-F. Yeh, C. Puhrsch, D. Pollack, D. Genzel, D. Greenberg, E. Z. Yang, J. Lian, J. Mahadeokar, J. Hwang, J. Chen, P. Goldsborough, P. Roy, S. Narenthiran, S. Watanabe, S. Chintala, V. Quenneville-Bélair, and Y. Shi, "Torchaudio: Building blocks for audio and speech processing," *arXiv preprint arXiv:2110.15018*, 2021.

[12] M. Shah, N. Pujara, K. Mangaroliya, L. Gohil, T. Vyas, and S. Degadwala, "Music genre classification using deep learning," in *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 974–978, 2022.

[13] M. Moerel, F. De Martino, K. Uğurbil, E. Yacoub, and E. Formisano, "Processing complexity increases in superficial layers of human primary auditory cortex," *Scientific Reports*, vol. 9, 04 2019.

[14] E. M. Parker and R. A. Sweet, "Stereological assessments of neuronal pathology in auditory cortex in schizophrenia," *Frontiers in Neuroanatomy*, vol. 11, 2018.

[15] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017.

[16] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," 2019.

[17] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," 2022.

[18] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," 2018.

[19] Z. Jackson, "Spoken$_d igit$,$''$ 2016.