



Degramnet: effective audio analysis based on a fully learnable time–frequency representation

Pasquale Foggia¹ · Antonio Greco¹ · Antonio Roberto¹ · Alessia Saggese¹ · Mario Vento¹

Received: 31 October 2022 / Accepted: 30 June 2023 / Published online: 19 July 2023
© The Author(s) 2023

Abstract

Current state-of-the-art audio analysis algorithms based on deep learning rely on hand-crafted Spectrogram-like audio representations, that are more compact than descriptors obtained from the raw waveform; the latter are, in turn, far from achieving good generalization capabilities when few data are available for the training. However, Spectrogram-like representations have two main limitations: (1) The parameters of the filters are defined a priori, regardless of the specific audio analysis task; (2) such representations do not perform any denoising operation on the audio signal, neither in the time domain nor in the frequency domain. To overcome these limitations, we propose a new general-purpose convolutional architecture for audio analysis tasks that we call DEGramNet, which is trained with audio samples described with a novel, compact and learnable time–frequency representation that we call DEGram. The proposed representation is fully trainable: Indeed, it is able to learn the frequencies of interest for the specific audio analysis task; in addition, it performs denoising through a custom time–frequency attention module, which amplifies the frequency and time components in which the sound is actually located. It implies that the proposed representation can be easily adapted to the specific problem at hands, for instance giving more importance to the voice frequencies when the network needs to be used for speaker recognition. DEGramNet achieved state-of-the-art performance on the VGGSound dataset (for Sound Event Classification) and comparable accuracy with a complex and special-purpose approach based on network architecture search over the Vox-Celeb dataset (for Speaker Identification). Moreover, we demonstrate that DEGram allows to achieve high accuracy with lightweight neural networks that can be used in real-time on embedded systems, making the solution suitable for Cognitive Robotics applications.

Keywords Deep learning · Audio representation learning · Signal processing · Sound event classification · Speaker identification

1 Introduction

In the last years, Deep Learning models achieved impressive performance in several machine learning tasks, ranging from scene understanding to text translation. The crucial features that allowed these models to outperform classical statistical methods are their ability to learn discriminant features directly from raw data.

Nowadays, this trend has been also confirmed in the field of audio analysis, in which end-to-end Convolutional

Neural Network (CNN) architectures have been proposed for the task of audio classification [23, 24, 27, 33]. These models take as input the raw waveform and stack several layers of strided-convolutions and max-pooling to create a compact representation given as input to the classification layers. However, raw waveform CNNs can be affected by overfitting and need huge datasets to generalize well on unseen data [29]. Fixed the CNN architecture and the input length, the common increase in the sampling rate, required to obtain a better resolution of the frequency spectrum, further amplifies the above mentioned problem.

In contrast with end-to-end approach, the scientific community demonstrated the effectiveness of Spectrogram-like [3] representations (e.g., Mel Spectrogram [8], Gammatonegram [19]) combined with 1D and 2D CNN

✉ Antonio Greco
agreco@unisa.it

¹ University of Salerno, Via Giovanni Paolo II 132, Fisciano, (SA), Italy

architectures [20, 24, 14] [2]. These are time–frequency representations of the input signal and are typically adopted in order to decrease the input size and, thus, to reduce the probability of overfitting. Moreover, once fixed the number of filters, the input shape does not vary with the sampling rate; this allows to overcome the issues which affect the raw waveform models. Finally, combining these representations with models that avoid strides and temporal poolings, it is possible to simply extract both temporal and frequency descriptors from the signal. These features determined the success of Spectrogram-like representations in all those applications in which the time information is dominant, such as Automatic Speech Recognition [1], Sound Event Detection [6] and Source Separation [25]. This is also confirmed by the results of the last DCASE challenge [13]: Indeed, for each of the six competitions, the model which ranked first was based on spectrogram-like features.

Even if the Spectrogram-like representations are compact and effective, they still rely on some fixed and pre-designed filter banks [30]. On one hand, the additional hyperparameters of these filters must be manually tuned; on the other hand, the use of hand-crafted features allows to avoid overfitting on small datasets, generalizing well with unseen audio samples [12].

In order to reduce the problem of overfitting in end-to-end models and to explicitly represent frequency information, Ravanelli et al. [31] proposed a neural network, called SincNet, able to learn from raw data the frequencies of interest in the first convolutional layer by using a trainable sinc filter bank. Differently from standard convolutional layers, this technique only needs to learn the cutoff frequencies of the band-pass filters. In this way, the resolution of the filter (i.e., its size) can be extended without increasing the number of parameters to train. Furthermore, the band-pass filter bank has the capability to filter out the parts of the spectrum not useful for the classification, focusing only on the frequencies-of-interest. Although SincNet is able to reduce the number of parameters to learn in the first convolutional layer, the remaining part of the network (following convolutional / dense layers) still works on raw (filtered) waveforms and needs a large amount of data to obtain good generalization capabilities. Also, it cannot dynamically perform signal denoising in presence of overlapping background noise over the learned frequency components, resulting in a drop in terms of performance when working in uncontrolled environments.

To address this problem, we have recently proposed a Denoising-Enhancement Layer (DELayer) [15], that allows to dynamically weight each frequency band with an attention mechanism and, consequentially, to increase the robustness of the model to noise. The DELayer proved to be more reliable in extreme conditions, gaining around 5%

in term of recall at very low signal-to-noise ratio (5 dB) with respect to the baseline architecture on the problem of Sound Event Detection. However, the network only takes into account the frequency components, while the temporal evolution of the audio signal is not considered. Therefore, the DELayer is not able to remove background noise limited in time and may assign lower weights to some frequency bands even if not needed. Moreover, the attention mechanism is not cost-less; in fact, the full neural network needs to batch multiple audio frames to run in real-time on CPU based architectures.

At the light of the above considerations, in this paper we propose a novel trainable time–frequency representation, namely DEGram, able to autonomously adapts itself to the specific problem at hands. It achieves robustness with respect to environmental noise thanks to the introduction of a joint time–frequency attention mechanism. The proposed representation is the result of two consecutive layers, namely SincGram and TF-DELayer (Time–Frequency DELayer).

SincGram (1) computes a band-pass filtered spectrogram of the input signal with the possibility to train the filters in an end-to-end manner. To this aim, the proposed layer takes as input the spectrogram of the input signal and performs the filtering as a matrix multiplication, which corresponds to a convolution in the time domain. Each column in the matrix of the weights corresponds to an approximated rectangular window which carries out a band-pass filtering. We approximated the rectangular window with the well-known Butterworth window [5] to make the function differentiable and, therefore, to favor the learning of the weights. It is worth pointing out that also SincNet approximates the sinc filters in the first layer, which are infinite in the time domain but modeled by a finite convolutional kernel.

Our SincGram inherits all the advantages of the Spectrogram: (1) A receptive field which is invariant to the sampling rate because of the fact that the Short-Time Fourier Transform (STFT) uses a fixed-in-time window to frame the audio signal; (2) an explicit representation of both the frequency and temporal components; (3) a frame-based temporal resolution which fosters the propagation of high-level temporal features in convolutional architectures. Furthermore, being a Spectrogram-like representation, SincGram overcomes the disadvantages of end-to-end models while achieving more flexibility thanks to learnable filters.

The output of SincGram is an intermediate time–frequency representation which is passed to the TF-DELayer. The TF-DELayer adds some important and not negligible improvements over the attention mechanism. We adapted the Spatial and Channel Squeeze and Excitation attention [32] to the time–frequency domain obtaining a triple

improvement: (1) squeezing the input before computing the attention coefficient, we reduce the computational complexity of the model; (2) computing the attention weights both in time and frequency, we improve the capability of the representation to filter out background noise; (3) applying the attention function on a squeezed version of the spectrogram, we drastically reduce the overhead of the parameters and the risk of overfitting.

We called the denoised and enhanced representation in output from the TF-DELayer, DEGram. Being a 2D signal, DEGram can be combined with all the 2D convolutional architectures widely adopted in the literature in order to address problems such as model bias and computational complexity. In particular, in this paper we combine DEGram with ECANet [35] to develop DEGramNet, a general purpose CNN for audio analysis tasks. Compared to DENet, DEGramNet has been designed to address audio classification tasks instead of sound event detection ones, i.e., determining the start and the end of each audio event in the input signal.

We proved the adaptability and the robustness of DEGramNet over two audio classification tasks: Sound Event Classification and Speaker Identification. In particular, we compared DEGramNet with several state-of-the-art CNNs over two widely adopted datasets, namely VGGSound [7] and VoxCeleb1 [26], very popular benchmarks for audio classification in the wild (i.e., acquired in unconstrained environments and characterized by environmental noise). Furthermore, we conducted an ablation study on the effect of each step of the proposed representation and we visually show the validity of the design choices. Finally, we evaluated the latency of the proposed representation on two GPU-enabled devices, i.e., a server with the NVIDIA Titan XP GPU and the NVIDIA Jetson Xavier NX embedded system, proving that DEGram is able to reduce the processing time with respect to spectrogram-based CNNs, while improving the accuracy.

Therefore, we can summarize the contributions of this paper as follows:

- we propose *DEGramNet*, a novel general purpose audio CNN that can be easily adapted to different audio classification problems, and that demonstrated excellent performance in unconstrained environments over different tasks, namely Sound Event Classification and Speaker Identification;
- DEGramNet inherits its main features from the proposed *DEGram*, a novel fully learnable time–frequency audio representation in which the frequencies of interest are learned from the training data instead than being fixed as in standard Spectrogram-like representations; moreover, it is able, thanks to a time–frequency attention module, to efficiently amplify the time and

frequency components in which the audio signal is actually localized and attenuate the others.

The paper is structured as follows: in Sect. 2 the various steps for obtaining the proposed representation is explained in detail and the DEGramNet architecture is described. In Sect. 3 we describe the experimental framework, namely the datasets, the metrics and some details about the training procedure, while in Sect. 4 we report the results, discussing the main findings and the insights inferred from the ablation study. Finally, in Sect. 5, we draw the conclusions.

2 Proposed method

The architecture of the proposed method is depicted in Fig. 1. In the first layer, the spectrogram is extracted from the input audio signal. Then, the DEGram representation allows to select the more relevant frequency bands from the spectrogram through the SincGram learnable filters; afterward, for each audio sample the attention mechanisms attenuate the additive background noise and amplify the useful time–frequency components (more details are reported in Sect. 2.1). Once the time–frequency representation has been computed, standard Convolutional Neural Networks and/or Recurrent Neural Networks can be used for addressing a specific audio analysis task. In the proposed DEGramNet (detailed in Sect. 2.2) we chose the ECA-Net architecture [35] as CNN for the audio classification task.

2.1 DEGram: denoising-enhancement spectrogram

The DEGram representation results from two layers: the former, namely SincGram, learns from the raw data the relevant frequency bands for the audio analysis task at hand; the latter, namely TF-DELayer, applies a time–frequency attention module to dynamically remove additive background noise and enhance those time–frequency components of the input representation useful for the classification.

2.1.1 SincGram

The SincGram layer is composed of M band-pass filters that can be learned during the training process and are represented in the frequency domain. In this way, the layer can learn which are the relevant frequency bands for the specific audio analysis task. It is known that a perfect band-pass filter corresponds to a rectangular window in the frequency domain, which is a not differentiable function; therefore, it cannot be optimized through gradient-based

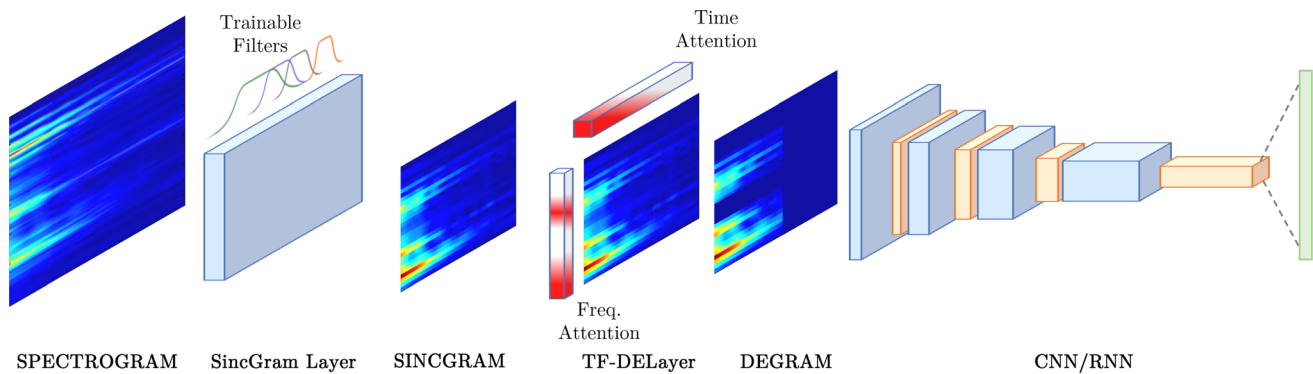


Fig. 1 DEGram-based CNN architecture, audio representations are labeled in uppercase and layers in lowercase. Computed the spectrogram of the audio signal, the SincGram representation is then obtained by filtering the spectrogram through trainable band-pass filters (SincGram Layer). The DEGram representation is finally

computed by applying a temporal and frequency attention layer able to focus on the relevant features of the input audio signal. Upon the learned audio representation, all the state-of-the-art CNNs and RNNs can be used in order to deal with different audio analysis tasks

approaches. To overcome this problem, we approximated the rectangular window with the Butterworth window [5].

The resulting equation of our filters is thus:

$$g(f, \alpha, \beta, n) = \frac{1}{\sqrt{1 + \left(\frac{f-\alpha}{\beta}\right)^{2n}}} \quad (1)$$

where g is the impulsive response of the filter centered at frequency α , and having scaling factor β and order n ; f represents the frequency domain independent variable. Figure 2 shows how an increasing window order results in a better approximation of the rectangular window. On the other hand, a more shaped window results in more flat gradients and then it suffers of the vanishing gradient problem.

Considering that the filtering operation can be computed as a multiplication in the frequency domain, the SincGram layer can be formulated as follow:

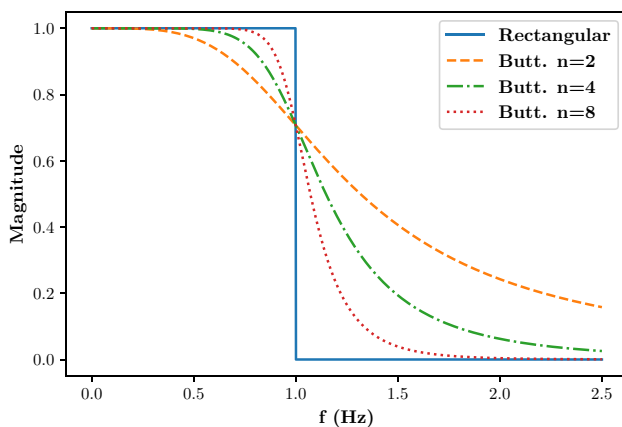


Fig. 2 Magnitude spectrum of the rectangular and Butterworth windows. The Butterworth window has been computed for three different orders, i.e., 2, 4, and 8. As the order increases, the Butterworth window better approximates the rectangular window

$$\text{SincGram}(X, \bar{\alpha}, \bar{\beta}) = X \cdot W(\bar{\alpha}, \bar{\beta}) \quad (2)$$

where $X \in \mathbb{R}^{T \times F}$ is the input spectrogram composed of T time steps and F frequency bins; $\bar{\alpha}, \bar{\beta} \in \mathbb{R}^M$ are the weights of the layer, whose size is equal to the number of filters to train; $W(\bar{\alpha}, \bar{\beta}) \in \mathbb{R}^{F \times M}$ is the resulting filtering matrix, in which each column represents a band-pass filter computed over the frequency bins of the spectrogram. For the sake of readability, the dependency between the SincGram layer and the order n of its filters is not reported because the order is an hyperparameter and, thus, it is not learned during the training procedure.

Differently from standard fully connected and convolutional layers, the proposed SincGram only needs to learn the center frequencies and the scaling factors of the filters, considerably reducing the number of parameters to optimize. Being a linear transformation of the spectrogram (as shown in Eq. 2), SincGram inherits a receptive field invariant to the sampling rate (unlike the raw waveform models) due to the fact that the STFT frames the audio signal through a fixed-duration window. When the sampling rate increases, it is common to increase the number of frequency bins F to balance the frequency resolution of the spectrogram itself; this operation results into a heavier network when only the spectrogram is used, since the feature maps are bigger. On the other hand, the resolution of the energies computed by SincGram can be extended without changing the input shape, which is only function of the time steps T and the number of filters M .

Finally, SincGram explicitly represents the time and frequency components of the input signal. In this way, the output representation is well suited for those tasks in which there is the need to temporally align the inputs and the outputs of the neural network architecture (e.g., audio event detection and speech recognition).

2.1.2 TF-DELayer

Even if SincGram is able to learn the relevant frequencies for the problem at hand, it cannot deal with additive noise in those components, which are common in real noisy environments. In fact, the weight associated to each frequency band is fixed by the following layers within the neural network architecture. To address this problem, we proposed in [15] DELayer, an attention mechanism built on top of the SincNet output which proved to dynamically filter background noise while enhancing information-full bands for the classification. Although the existing DELayer allows to reach good performance, the need of an auxiliary neural network to compute the band weights entangles the optimization and slows down the model inference. Furthermore, the attention mechanism dynamically identifies only the frequency bands with higher importance without any consideration of the temporal information.

For these reasons, we propose the Time Frequency DELayer (Fig. 3) which combines time and frequency attention maps to enhance those useful frequency bands only in the time slots in which the sound of interest is located. Moreover, the improved layer computes the attention weights taking into account all the frequency bands together, in order to better identify the most relevant ones.

the TF-DELayer adapts the spatial and channel two-dimensional attention [32] to a time–frequency mono-dimensional representation; therefore, it consists of two parallel branches for frequency and time attention, respectively.

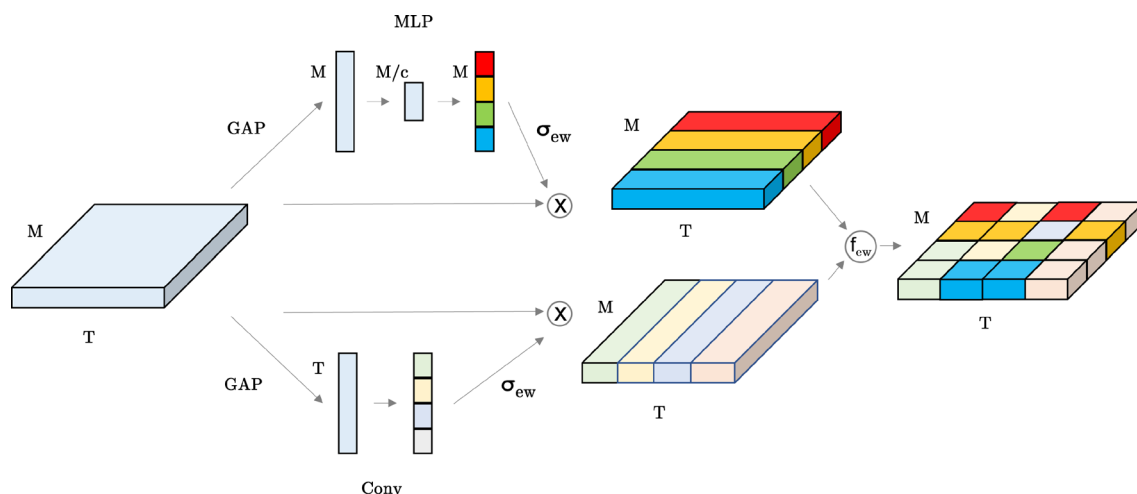


Fig. 3 TF-DELayer computations on the input representation composed by T time steps and M filters. The temporal and frequency attention are independently computed. The frequency attention scores are computed by passing the squeezed representation along the time dimension to a bottleneck MLP. The temporal attention is computed in the same way but replacing the bottleneck MLP with a

The frequency attention branch can be formalized as follows:

$$Freq_{att}(s) = \sigma_{ew}(MLP(GAP(s))) \times s \quad (3)$$

It squeezes the input SincGram s along the time component by aggregating the information of each frequency band through a Global Average Pooling (GAP) layer, to reduce the complexity of the representation. Then, a bottleneck multilayer perceptron (MLP) is used to compute the attention coefficients. It is characterized by a reduction coefficient c , which allows to further reduce the number of parameters. The attention coefficients are then projected in the range $[0, 1]$ through the element-wise sigmoid function σ_{ew} . Finally, the filtered frequency bands are computed by multiplying the resulting attention coefficient with the original input. It is important to point out that the MLP can be used thanks to the fact that the number of filters in the SincGram representation is fixed and does not depend on the input length and the sampling rate.

With respect with the previous DELayer, the TF-DELayer considerably reduces the computational complexity of the frequency attention branch thanks to the squeezing operation introduced by the GAP layer. This operation not only speeds up the training and the inference time, but also reduces the number of parameters to train, making the optimization easier and allowing the model to better deal with small datasets. Moreover, the way in which the frequency bands to enhance are identified is *orthogonal* with respect to the previous layer in which each component is analyzed independently; in the new layer, the GAP layer

Convolutional layer to manage the variable length of the input signal and squeezing the frequency dimension. The sigmoid function is applied to normalize the obtained scores in the range $[0,1]$. Finally, the scaled temporal and frequency features are combined in the DEGram representation by adopting an element wise aggregation function (e.g., max and average)

exploits the inter-correlations between the frequency bands to better identify the most relevant ones.

The time attention mechanism can be formalized as follows:

$$Time_{att}(s) = \sigma_{ew}(Conv1D(GAP(s))) \times s \quad (4)$$

Also this module squeezes the GAP of the input SincGram s to reduce the complexity of the layer itself but, in this case, the aggregation is computed at frequency level. Since different audio samples can have variable length, the attention maps are computed through a mono-dimensional convolutional layer (Conv1D) followed by a sigmoid activation function. A padding is applied to the input in order to obtain a number of attention weights equivalent to the time steps of the input. The time-filtered representation is computed as a multiplication between the attention weights and the input SincGram. Adding this attention branch, the TF-Layer not only enhances relevant frequency bands while attenuating background noise, but also focuses the attention only in that time slots in which the sound of interest is located.

The final DEGram representation is computed starting from the two filtered representations through an element-wise aggregation function f :

$$DEGram(s) = f_{ew}(Freq_{att}(s), Time_{att}(s)) \quad (5)$$

2.2 DEGramNet

Having available the low-level representation of the audio signal, we need to compute high-level features to address a particular audio analysis task. As shown in Fig. 1 and discussed in Sect. 2.1, the proposed DEGram is a 2-dimensional time–frequency representation and; thus, it can be combined with all the well-known Convolutional and Recurrent Neural Networks. This is a crucial feature of our method, which allows researchers to use the proposed learnable representation with any neural network architecture.

In our DEGramNet, we adopt the Efficient Channel Attention Network (ECA-Net) as reference CNN architecture to prove the effectiveness of our representation. ECA-Net [35] has been chosen for two main reasons. First, the network is composed of *Residual Blocks* [17] [18] which learn a residual function of the input in contrast to VGG-like architectures [34], as shown in Fig. 4. Residual blocks proved to be easier to optimize and, moreover, they allow to alleviate the problem of the vanishing gradient in very deep neural network. This effect is achieved thanks to the direct connection with the input signal which allows a direct back-propagation of the gradients themselves.

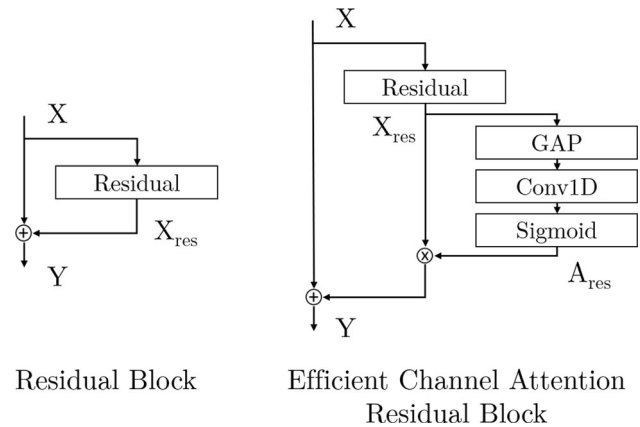


Fig. 4 Comparison between residual block and efficient channel attention residual block

Table 1 DEGramNet convolutional architecture. Each conv block corresponds to an ECA residual block. The global average pooling layer upon the convolutional architecture allows the network to deal with variable length audio samples

Layer name	Residual block convolutions
SincGram	–
DELayer v2.0	–
Conv1	7×7, 64, stride 2
maxpool1	3×3, stride 2
Conv2	$\begin{bmatrix} 1 \times 1 \text{ conv, } 64 \\ 3 \times 3 \text{ conv, } 64 \\ 1 \times 1 \text{ conv, } 256 \end{bmatrix} \times 3$
Conv3	$\begin{bmatrix} 1 \times 1 \text{ conv, } 128 \\ 3 \times 3 \text{ conv, } 128 \\ 1 \times 1 \text{ conv, } 512 \end{bmatrix} \times 8$
Conv4	$\begin{bmatrix} 1 \times 1 \text{ conv, } 256 \\ 3 \times 3 \text{ conv, } 256 \\ 1 \times 1 \text{ conv, } 1024 \end{bmatrix} \times 36$
Conv5	$\begin{bmatrix} 1 \times 1 \text{ conv, } 256 \\ 3 \times 3 \text{ conv, } 256 \\ 1 \times 1 \text{ conv, } 1024 \end{bmatrix} \times 3$
Average pool, n_classes-d fc, softmax	

The second feature of ECA-Net is the *Efficient Channel Attention module*. Differently from the TF-DELayer, the ECA module is applied on the feature maps (i.e., channels). In particular, it aims to improve the hidden representations produced by the neural network by explicitly modeling the inter-correlations between the feature maps in each convolutional layer. It has been proved that this channel-wise feature re-calibration allows to increase the generalization capabilities of the standard residual networks which are commonly adopted for audio analysis tasks [21, 32, 35].

From the computational point of view, the ECA module works similarly to the frequency attention mechanism

reported in Eq. 3, but the inter-correlations between the different feature maps are computed locally through the use of a 1-dimensional convolutional layer instead of the bottleneck MLP. Even if the choice of replacing the MLP with a convolutional layer implies a sort of locality of the inter-correlations between the feature maps, it allows to efficiently compute the attention weights and to better scale with the network depth, avoiding the explosion of the model complexity.

The kernel size k of the convolutional filter is computed as follows:

$$k = \left\lceil \frac{\log_2(C) + 1}{2} \right\rceil_{\text{odd}} \quad (6)$$

where C is the number of feature maps. This choice helps to reduce the number of hyperparameters to tune.

The final architecture of DEGramNet is summarized in Table 1.

3 Experimental framework

In order to prove the generalization capability of DEGramNet and to validate the assumptions done for the proposed DEGram representation, we performed experiments on two audio classification tasks characterized by different sound of interest: Sound Event Classification and Speaker Identification. In particular, in order to benchmark the robustness of the proposed DEGramNet to environmental noise, we adopted two large-scale dataset acquired in unconstrained conditions: VGGSound [7], for the task of Sound Event Classification [16], and VoxCeleb1 [26], for the task of Speaker Identification [11]. In this section, we describe the datasets (Sect. 3.1), the metrics adopted to evaluate the performance of DEGramNet (Sect. 3.2), and the implementation details necessary to reproduce our experimental framework (Sect. 3.3).

3.1 Datasets

3.1.1 VGGSound

VGGSound [7] is a large-scale audio-visual dataset recently proposed for the task of sound event classification in the wild, i.e., in uncontrolled conditions. For each sample, the object emitting the sound is visible in the corresponding video. VGGSound contains 309 classes of sound events ranging from people sounds to nature and animals. The dataset is composed of about 200k video clips downloaded from YouTube and acquired in the wild, allowing to achieve a two-fold result: a substantial increase in the task difficulty due to different challenging acoustic environment and noisy acquisitions and the removal of the

bias due to the acquisition conditions (e.g., the recording device).

VGGSound has a pre-defined training-test set division and contains audio samples that are 10-s long. The audio samples in the test set are uniformly distributed with respect to the sound classes, while the training set is highly unbalanced.

3.1.2 VoxCeleb1

VoxCeleb1 [26] is a large-scale audio-visual speech dataset for the tasks of Speaker Identification and Speaker Verification. The dataset consists of more than 100 k utterances acquired from 1215 celebrities in the wild. Even in this case, VoxCeleb1 has been acquired starting from YouTube video clips and, thus, contains a wide range of recording conditions, ranging from outdoor stadiums to quiet studio interviews. The speakers within VoxCeleb1 are characterized by a wide range of accents, ages and ethnicity; moreover, the dataset is gender-balanced, with 55% of male speakers. The summary of the dataset is reported in Table 2.

VoxCeleb1 has a pre-defined training-validation-test set division; all the sets are unbalanced. The utterances within the dataset have a variable duration, ranging from less than 1–20 s.

3.2 Metrics

To evaluate the performance of DEGramNet, we adopted the evaluation protocol proposed in the papers of the datasets considered in our experimental analysis [7, 26]. The metrics used for the evaluation of the accuracy include mean Average Precision (mAP), macro-averaged Area Under the receiver operating characteristic (ROC) Curve (AUC), the equivalent d-prime class separation index, Top-1 Accuracy, and Top-5 Accuracy. However, for the VoxCeleb1 dataset, only Top-1 and Top-5 Accuracy scores were computed.

The mAP score (Eq. 7) is the mean of the Average Precision (AP) scores computed for each class.

Table 2 Statistics of the VoxCeleb1 dataset for speaker identification. fields characterized by three value represents max/avg/min statistics respectively

# of POIs	1215
# of male POIs	690
# of videos per POI	36 / 18 / 8
# of utterances per POI	250 / 123 / 45
Length of utterances (s)	145.0 / 8.2 / 4.0

$$mAP = \frac{1}{N} \sum_i^N AP_i \quad (7)$$

The AP [4] is, in turn, the average percentage of correctly detected samples within the ranked list of the detections by varying the size of the list (from 1 to the number of detections).

The AUC [10], as suggested by the name, is computed as the area under the ROC curve, which shows the True Positive Rate and False Positive Rate as functions of the detection threshold. The AUC score is calculated for each class and then averaged in order to obtain a priors-invariant metric score.

The relative d-prime index d' represents the separability between classes and is computed as:

$$d' = \sqrt{2}F^{-1}(AUC) \quad (8)$$

where F^{-1} is the inverse cumulative distribution function for a Gaussian unit.

Finally, the Top-1 and Top-5 Accuracy scores are obtained by computing the percentage of correctly classified samples considering the top one and the top five classes with the highest predicted probability, respectively.

3.3 Implementation details

The DEGramNet architecture has been trained through the Adam optimization algorithm. The learning rate used to train the neural network is equal to $3e-4$, while the batches consist of 64 samples. These hyperparameters have been selected with a Bayesian Optimization approach. During the training procedure, the learning rate has been reduced of a factor 10 if the accuracy on the validation set did not decrease for 7 consecutive epochs, in order to avoid the presence of a plateau in the error function. Moreover, an Early Stopping strategy has been adopted to not overfit the training data; the patience has been set to 10. To avoid the specialization of the network over the most represented classes, we balanced the dataset at the beginning of each epoch by randomly under-sampling the majority classes without replacement.

We set the order of the Butterworth window within the SincGram representation to 4 while the compression coefficient is set to 0.0625. Finally, we fixed the number of trainable filters to 64. These hyperparameters have been optimized through a grid search, while the number of spectrogram bins has been set to 257 as done in [7].

We adopted different regularization strategies within the training of DEGramNet. In order to obtain a neural network able to deal with audio samples of different length, we created at each training iteration a batch with time length randomly chosen in the range [4, 6] seconds. To this aim,

we replicated the samples shorter than the selected window and cut the longer ones. Finally, we augmented the training set using the SpecAugment procedure [28]. It is worth mentioning that the augmentation is applied within the network itself after the computation of the DEGram representation to better regularize the remaining part of the CNN. In Table 3 we show that the training of the proposed DEgramNet converges in a few hours on quite big datasets such as VGGSound and VoxCeleb1. By using a NVIDIA Titan Xp GPU, the training requires 56 epochs and 8 h on VGGSound, while on VoxCeleb1 it takes 51 epochs and 27 h. This result confirms the efficiency of the proposed learning procedure, while the experimental analysis reported in the following section will demonstrate its effectiveness.

4 Experimental results

In this section, we report the experimental results obtained on VGGSound and VoxCeleb1 by the proposed DEGramNet, compared with the ones achieved by other state-of-the-art algorithms optimized for the same tasks (Sect. 4.1). Furthermore, we carried out an ablation study on the contribution of each additional step of the proposed representation (i.e., SincGram and DEGram) considering, for the sake of comparability with other methods in the literature, the well-known ResNet architecture [17, 18] (Sect. 4.2). Moreover, an in-depth analysis of the learned representation has been conducted by visually comparing the input obtained with Spectrogram, SincGram and DEGram. Finally, the processing time has been evaluated on a server with the NVIDIA Titan XP GPU and the NVIDIA Jetson Xavier NX embedded system (Sect. 4.3) to investigate the usability of the proposed representation in real applications.

4.1 Results

We report in Tables 4 and 5 the results obtained by DEGramNet on VGGSound and VoxCeleb1. We compared the proposed model with other methods evaluated on the

Table 3 Statistics about the convergence of the learning process of DEgramNet on VGGSound and VoxCeleb1 in terms of number of epochs, validation loss and training time. The training has been done on a NVIDIA Titan Xp GPU

Dataset	Epochs	Val. loss	Training time (h)
VGGSound	56	1.691	8
VoxCeleb1	51	1.099	27

same datasets. These results have been reported from [7, 9, 22, 26] in order to make a fair comparison.

DEGramNet achieved state-of-the-art performance on the VGGSound dataset with a mAP of 0.547 and a Top-1 accuracy equal to 0.526. The proposed method outperformed all the standard CNN architectures for audio analysis based on Residual Networks. DEGramNet gained 0.4% in terms of AUC and 0.083 in terms of d-prime compared to the deeper ResNet (i.e., ResNet50, with AUC and d-prime equal 0.973 and 2.735, respectively), demonstrating a strong capability to distinguish positive and negative samples despite the high variability of the dataset. The gap increases up to 0.9% and 0.191 for smaller architectures as ResNet18. The proposed method also showed better ranking performance, gaining from 1.7% (w.r.t. ResNet50) to 3.5% (w.r.t. ResNet18) in terms of Top-5 accuracy. The result is further confirmed by the mAP metric, which considers the overall ranking performance and not only the Top-5 predictions. In this case, DEGramNet gained from 1.5% if compared to ResNet50 (0.532), up to 3.1% with respect to ResNet18 (0.516).

DEGramNet outperformed the SlowFast architecture, the method achieving so far the best performance on the VGGSound test set, on four metrics over five (except for Top-5 Accuracy). Although SlowFast performs a multi-stream analysis over the Mel-Spectrogram representation and requires a complex architectural design, it was outperformed by DEGramNet on the Top-1 accuracy (gaining 0.1%), AUC and d-prime scores (gaining 0.3% and 0.057, respectively). Therefore, these results prove that the proposed CNN is slightly more accurate on that task by only learning an effective audio representation and without any additional design effort. Moreover, even if DEGramNet obtained a lower Top-5 accuracy score (− 1%), it proved to better rank classes in general by obtaining a higher mAP score (0.3%).

Looking at the results obtained on the VoxCeleb1 dataset, the outcomes are quite similar. Even in this case, DEGramNet outperformed all the standard CNN architectures in speaker identification, obtaining a Top-1 Accuracy

of 0.866. In particular, the proposed network gained 5.5% with respect to ResNet34 (0.813) and 6.1% if compared with VGG-like neural networks (0.805). Moreover, we compared our results with AutoSpeech, which is a CNN architecture specifically optimized with a Network Architecture Search (NAS) approach on the VoxCeleb1 dataset for the task of Speaker Identification and Verification. We can observe that DEGramNet obtained comparable results to this special-purpose method without any effort, either hand-crafted or computational, in the design step; therefore, our method demonstrated to be a general purpose architecture for audio analysis tasks. Indeed, this result proves the adaptability of DEGram to different tasks, being the representation able to learn in a single training step the frequencies bands of interest for the specific problem at hand.

4.2 Ablation study

In order to further analyze the positive contribution of the proposed representation with respect to the standard Spectrogram, we conducted an ablation study to evaluate the impact of the single design choices. In particular, we performed a quantitative and qualitative analysis by firstly removing the attention mechanism and then the SincGram filtering layer. We also performed experiments with a representation combining Spectrogram and DELayer, hereinafter called DESpectrogram; this variant allows to measure the positive contribution of SincGram to the proposed representation. In order to make a comparison independent from the specific dataset chosen for training the methods, we compared the results obtained by the four representations given in input to the ResNet-18 model on the VGGSound and VoxCeleb1 test sets. We reported the results of existing spectrogram-based CNNs from the literature for a fair comparison [7, 9]. The obtained results are shown in Tables 6 and 7.

Table 4 Comparison of the results achieved by DEGramNet and other state-of-the-art methods on the VGGSound test set (Sound Event Classification) in terms of mAP, AuC, d-prime, Top-1 and Top-5 Accuracy. The best results for each performance metric are highlighted in bold

Method	References	mAP	AUC	d-prime	Top-1 accuracy	Top-5 accuracy
ResNet-18	[7]	0.516	0.968	2.627	0.488	0.746
ResNet-34	[7]	0.529	0.972	2.703	0.505	0.758
ResNet-50	[7]	0.532	0.973	2.735	0.510	0.764
SlowFast	[22]	0.544	0.974	2.761	0.525	0.781
DEGramNet	(Our)	0.547	0.977	2.818	0.526	0.774

Table 5 Comparison of the results achieved by DEGramNet and other state-of-the-art methods on the VoxCeleb1 test set (Speaker Identification) in terms of Top-1 and Top-5 Accuracy. The best results are highlighted in bold

Method	References	Top-1 accuracy	Top-5 accuracy
ResNet-18	[9]	0.795	0.910
VGG-M	[26]	0.805	0.921
ResNet-34	[9]	0.813	0.945
DEGramNet	(Our)	0.866	0.950
AutoSpeech (NAS)	[9]	0.876	0.960

Table 6 Comparison of the results obtained on the VGGSound test set by ResNet18 trained with Spectrogram, SincGram, DESpectrogram and DEGram. The best results for each performance metric and

for each CNN are highlighted in bold. The CNNs trained with the proposed representations always outperform the corresponding architecture trained with Spectrogram

Method	References	mAP	AUC	d-prime	Top-1 accuracy	Top-5 accuracy
ResNet-18	[7]	0.516	0.968	2.627	0.488	0.746
SincGram-ResNet18	(Our)	0.519	0.974	2.747	0.496	0.753
DESpectrogram-ResNet18	(Our)	0.507	0.972	2.706	0.470	0.740
DEGram-ResNet18	(Our)	0.526	0.975	2.765	0.507	0.763

Table 7 Comparison of the results obtained on the VoxCeleb1 test set by ResNet18 trained with Spectrogram, SincGram, DESpectrogram and DEGram. The best results for each performance metric and

for each CNN are highlighted in bold. The CNNs trained with the proposed representations always outperform the corresponding architecture trained with Spectrogram

Method	References	Top-1 Accuracy	Top-5 Accuracy
ResNet-18	[9]	0.795	0.910
Sincgram-ResNet18	(Our)	0.826	0.934
DESpectrogram-ResNet18	(Our)	0.827	0.932
DEGram-ResNet18	(Our)	0.831	0.939

We can immediately note that, regardless of the task we are dealing with, the proposed representations allow to substantially improve the performance. The spectrogram-based ResNet18 achieved a Top-1 Accuracy of 0.488 and 0.795 on the VGGSound and VoxCeleb1 datasets. Adding to the network the capability to learn the frequencies of interest (SincGram), the CNN was able to improve the performance of around 1% and 3%; this result demonstrates the positive effect of the introduction of the SincGram layer in the neural network. A similar improvement can be seen on the VoxCeleb dataset with the DESpectrogram representation, able to better deal with the noise characterizing the dataset. On the other hand, the DESpectrogram representation obtains the worst result on VGGSound. A possible interpretation of this result is that the TF-DELayer is not able alone to distinguish the frequencies with overlapping noise from those of interest for all the considered sounds (i.e., 309 classes). Interestingly, the accuracy increases of further 1% for both the tasks when we use DEGram, combining the SincGram

representation (learnable frequencies of interest) and the TF-DELayer (noise robustness). This results confirm the effectiveness of the proposed architecture.

To deduce further insights, we performed a qualitative analysis on the various representations, reported in Fig. 5. In particular, we visually compared the Spectrogram, SincGram and DEGram representations of a noisy speech audio sample, in order to identify the key elements that allow the proposed solutions to be more effective. We can note that SincGram stretches the frequencies in the lower part of the spectrum, in which most of the relevant speech information is located. This result confirms the capability of the SincGram layer to learn a time-frequency representation in which the more relevant components of the sound (for the specific problem at hand) are emphasized. Nevertheless, as expected, also overlapping noise is amplified when the spectrum is processed by the learned filters. The DEGram representation mitigates this issue, since it is able to perform denoising through the temporal attention module. This result is evident by observing that

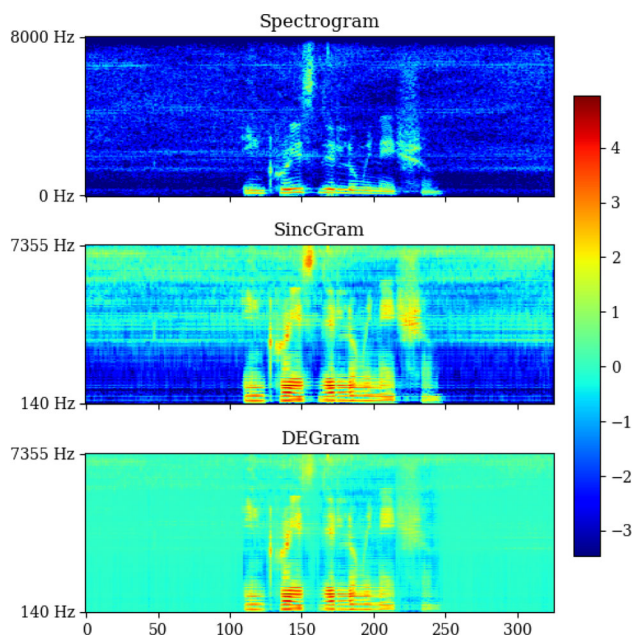


Fig. 5 Comparison between Spectrogram, SincGram and DEGram applied to a noisy speech sample. SincGram focuses on the most important frequency components (the lower part of the spectrum) and the effect is evident with respect to Spectrogram. Then, DEGram performs denoising on the frequencies affected by overlapping noise (higher part of the learned spectrum) and focuses on the time slots in which the sound of interest is located

all the components not overlapping with the speech signal (temporally centered) are close to 0 (teal colored). Moreover, looking at the upper part of the DEGram representation, we can note smoothed energies with respect to SincGram, proving that the frequency attention mechanism attempted to reduce the noise in that part of the spectrum.

The qualitative analysis confirms the assumptions made in the design of DEGram. The proposed audio

representation is actually able to amplify the relevant components of the spectrum and to attenuate the ones affected by noise, focusing only on the temporal steps in which the sound is located.

4.3 Processing time evaluation

To evaluate the computational impact of the proposed representation on the processing time of an audio analysis system, we compared the latency of the ResNet18 architecture with and without the proposed DEGram representation and the latency of a Spectrogram-based ResNet34. The evaluation has been carried out on two different GPU enabled devices, namely a server equipped with the NVIDIA Titan XP GPU and a NVIDIA Jetson Xavier NX, an embedded system for IoT and Robotics applications. In addition, we evaluated the processing time on the latter device with and without the use of the GPU. The inference time has been computed by averaging the processing time over 100 network predictions. The obtained results are reported in Table 8.

We can note that the proposed DEGram representation has a negligible overhead on GPU architectures (1 ms on the Titan XP and less than 10 ms on the Jetson Xavier NX) when using the same CNN architecture for the inference. On the same devices, DEGram is able to save up to 37 ms with respect to the ResNet34 architecture; this is not negligible, especially if we consider, as shown in the previous section, that ResNet18 trained with DEGram (Tables 6 and 7) achieves better accuracy than ResNet34 trained with Spectrogram (Tables 4 and 5). The processing speed-up is even higher when the computation is done on a CPU. In this case, the ResNet architecture based on DEGram reduces the processing time up to 110 ms on 10 s inputs, thanks to the reduced number of frequency features (from

Table 8 Processing time in milliseconds required by ResNet18, DEGram-ResNet18 and ResNet34 to process audio samples with variable input length (1, 3, 5, 10 s) over NVIDIA Titan Xp GPU and NVIDIA Jetson Xavier NX. DEGram-ResNet18 is able to reduce the processing time with respect to the spectrogram-based ResNet34 (achieving lower performance). The proposed architecture reduces the latency of the same CNN architecture on CPU-based architectures

Device	Input length (s)	ResNet18	DEGram-ResNet18	ResNet34
NVIDIA Titan XP	1	6	7	10
	3	7	7	10
	5	7	8	10
	10	10	9	15
NVIDIA Jetson Xavier NX (GPU)	1	165	174	180
	3	172	174	182
	5	177	177	199
	10	201	207	244
NVIDIA Jetson Xavier NX (CPU)	1	220	212	284
	3	298	244	408
	5	354	278	499
	10	479	362	745

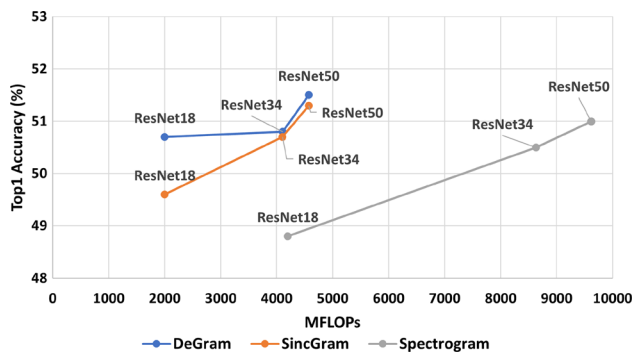


Fig. 6 Number of operations in MFLOPs required for an audio sample of 1 s by ResNet18, ResNet34 and ResNet50 with SincGram, DEGram and Spectrogram and accuracy (%) achieved by these neural networks. SincGram allows to halve the number of operations compared to the Spectrogram, with a substantial increase in accuracy. DEGram adds a negligible number of operations and achieves accuracy improvement

257 frequency bins to 64 trainable filters). The gap drastically increases comparing the proposed architecture with ResNet34, which requires around twice for computing sequences longer than 3 s. The result is further confirmed by counting the number of floating operations (FLOPs) required for a second of processed audio, reported in Fig. 6. ResNet18, ResNet34 and ResNet50 based on Spectrogram need 4.19, 8.64 and 9.62 GFLOPs, respectively. The proposed solution allows to halve the number of FLOPs for all the neural networks, while achieving a substantially higher accuracy. We can also note that DEGram adds a negligible amount of FLOPs, but it allows improve the accuracy, especially on a smaller architecture such as ResNet18.

5 Conclusions

In this paper we described and experimentally evaluated DEGramNet, a new CNN for audio analysis tasks which relies on the proposed SincGram and DEGram audio representations. The former is able to learn the frequencies of interest for the audio analysis problem we are dealing with, extracting task-specific features through trainable band-pass filters. The latter can denoise the input signal from environmental background noise using a time-frequency attention mechanism, focusing on the part of the input signal in which the sound of interest is temporally located and on the frequency components not affected by noise.

The experimental analysis demonstrated the effectiveness of DEGramNet, that was able to achieve state-of-the-art results on the VGGSound dataset (Sound Event Classification) and comparable results with a complex Network Architecture Search approach on the VoxCeleb1 dataset (Speaker Identification), proving to be a general purpose

architecture for audio analysis tasks. Moreover, our ablation study proved that the proposed DEGram representation was more effective than Spectrogram, being able to improve the performance of the same CNN architecture trained with both the representations and to achieve better accuracy with respect to deeper models. Finally, the reduction of the number of features computed by DEGram allows the same network to reduce the inference time on CPU architectures, which is a not negligible features in all those contexts that need real-time performance with strict computational constraints like, for instance, IoT and Cognitive Robotics applications.

Funding Open access funding provided by Università degli Studi di Salerno within the CRUI-CARE Agreement. The authors do not receive funding for this research.

Data availability The authors do not provide supplementary data and material.

Code availability The code will be available at: <https://github.com/MiviaLab/DEGramNet>.

Declarations

Conflicts of interest The authors declare no conflicts of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Abdel-Hamid O, Rahman Mohamed A, Jiang H, Deng L, Penn G, Yu D (2014) Convolutional neural networks for speech recognition. *IEEE/ACM Trans Audio Speech Lang Proc* 22(10):1533–1545
2. Al-Hattab YA, Zaki HF, Shafie AA (2021) Rethinking environmental sound classification using convolutional neural networks: optimized parameter tuning of single feature extraction. *Neural Comput Appl* 33(21):14495–14506
3. Allen J (1977) Short term spectral analysis, synthesis, and modification by discrete fourier transform. *IEEE Trans Acoust Speech Signal Proc* 25(3):235–238. <https://doi.org/10.1109/tassp.1977.1162950>
4. Buckley C, Voorhees EM (2004) Retrieval evaluation with incomplete information. In: *Proceedings of the 27th annual*

- international conference on Research and development in information retrieval - SIGIR '04, pp. 25–32. ACM Press
5. Butterworth S (1930) On the theory of filter amplifiers. *Exp Wirel Eng* 7(6):536–541
 6. Cakir E, Parascandolo G, Heittola T, Huttunen H, Virtanen T (2017) Convolutional recurrent neural networks for polyphonic sound event detection. *IEEE/ACM Trans Audio Speech Lang Proc* 25(6):1291–1303. <https://doi.org/10.1109/taslp.2017.2690575>
 7. Chen H, Xie W, Vedaldi A, Zisserman A (2020) Vggsound: A large-scale audio-visual dataset. In: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 721–725. IEEE, IEEE
 8. Davis S, Mermelstein P (1990) Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans Acoust Speech Signal Proc* 28(4):65–74. <https://doi.org/10.1016/b978-0-08-051584-7.50010-3>
 9. Ding S, Chen T, Gong X, Zha W, Wang Z (2020) AutoSpeech: Neural architecture search for speaker recognition. In: Interspeech 2020, pp. 916–920. ISCA. <https://doi.org/10.21437/interspeech.2020-1258>
 10. Fawcett T (2004) Roc graphs: notes and practical considerations for researchers. *Mach Learn* 31(1):1–38
 11. Foggia P, Greco A, Roberto A, Saggese A, Vento M (2023) Few-shot re-identification of the speaker by social robots. *Auton Robots* 47(2):181–192
 12. Foggia P, Saggese A, Strisciuglio N, Vento M, Petkov N (2015) Car crashes detection by audio analysis in crowded roads. In: 2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–6. IEEE, IEEE
 13. Font F, Mesaros A, Ellis DP, Fonseca E, Fuentes M, Elizalde B (2021) Proceedings of the 6th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2021). Barcelona, Spain
 14. Greco A, Petkov N, Saggese A, Vento M (2020) AREN: a deep learning approach for sound event recognition using a brain inspired representation. *IEEE Trans Inf Forensics Secur* 15:3610–3624. <https://doi.org/10.1109/tifs.2020.2994740>
 15. Greco A, Roberto A, Saggese A, Vento M (2021) DENet: a deep architecture for audio surveillance applications. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-020-05572-5>
 16. Greco A, Saggese A, Vento M, Vigilante V (2019) Sorennet: A novel deep network for audio surveillance applications. In: 2019 IEEE international conference on systems, man and cybernetics (SMC), pp. 546–551. IEEE
 17. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778. IEEE. <https://doi.org/10.1109/cvpr.2016.90>
 18. He K, Zhang X, Ren S, Sun J (2016) Identity mappings in deep residual networks. *Computer vision - ECCV 2016*. Springer, Cham, pp 630–645
 19. van Hengel PWJ, Krijnders JD (2014) A comparison of spectro-temporal representations of audio signals. *IEEE/ACM Trans Audio Speech Lang Proc* 22(2):303–313. <https://doi.org/10.1109/taslp.2013.2283105>
 20. Hershey S, Chaudhuri S, Ellis DPW, Gemmeke JF, Jansen A, Moore RC, Plakal M, Platt D, Saurous RA, Seybold B, Slaney M, Weiss RJ, Wilson K (2017) CNN architectures for large-scale audio classification. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 131–135. IEEE. <https://doi.org/10.1109/icassp.2017.7952132>
 21. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7132–7141
 22. Kazakos E, Nagrani A, Zisserman A, Damen D (2021) Slow-fast auditory streams for audio recognition. In: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 855–859. IEEE, IEEE. <https://doi.org/10.1109/icassp39728.2021.9413376>
 23. Kim T, Lee J, Nam J (2018) Sample-level CNN architectures for music auto-tagging using raw waveforms. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Neural Information Processing Systems (NIPS), IEEE. <https://doi.org/10.1109/icassp.2018.8462046>
 24. Kim T, Lee J, Nam J (2019) Comparison and analysis of SampleCNN architectures for audio classification. *IEEE J Sel Top Signal Proc* 13(2):285–297. <https://doi.org/10.1109/jstsp.2019.2909479>
 25. Lin KWE, Balamurali B, Koh E, Lui S, Herremans D (2020) Singing voice separation using a deep convolutional neural network trained by ideal binary mask and cross entropy. *Neural Comput Appl* 32(4):1037–1050
 26. Nagrani A, Chung JS, Zisserman A (2017) VoxCeleb: A large-scale speaker identification dataset. In: Interspeech 2017, pp. 2616–2620. ISCA. <https://doi.org/10.21437/interspeech.2017-950>
 27. Naranjo-Alcazar J, Perez-Castanos S, Martin-Morato I, Zucarello P, Ferri FJ, Cobos M (2020) A comparative analysis of residual block alternatives for end-to-end audio classification. *IEEE Access* 8:188875–188882. <https://doi.org/10.1109/access.2020.3031685>
 28. Park DS, Chan W, Zhang Y, Chiu CC, Zoph B, Cubuk ED, Le QV (2019) SpecAugment: A simple data augmentation method for automatic speech recognition. In: Interspeech 2019, pp. 2613–2617. ISCA
 29. Pons Puig J, Nieto Caballero O, Prockup M, Schmidt EM, Ehmann AF, Serra X (2018) End-to-end learning for music audio tagging at scale. In: Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018; 2018 Sep 23–27; Paris, France. p. 637–44. International Society for Music Information Retrieval (ISMIR)
 30. Purwins H, Li B, Virtanen T, Schluter J, Chang SY, Sainath T (2019) Deep learning for audio signal processing. *IEEE J Sel Top Signal Proc* 13(2):206–219. <https://doi.org/10.1109/jstsp.2019.2908700>
 31. Ravanelli M, Bengio Y (2018) Speaker recognition from raw waveform with SincNet. In: 2018 IEEE Spoken Language Technology Workshop (SLT), pp. 1021–1028. IEEE, IEEE. <https://doi.org/10.1109/slt.2018.8639585>
 32. Roy AG, Navab N, Wachinger C (2018) Concurrent spatial and channel ‘squeeze & excitation’ in fully convolutional networks. *Medical Image Computing and Computer Assisted Intervention - MICCAI 2018*. Springer, Cham, pp 421–429
 33. Saggese A, Strisciuglio N, Vento M, Petkov N (2016) Time-frequency analysis for audio event detection in real scenarios. In: 2016 13th IEEE international conference on advanced video and signal based surveillance (AVSS), pp. 438–443. IEEE
 34. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations
 35. Wang Q, Wu B, Zhu P, Li P, Zuo W, Hu Q (2020) ECA-net: Efficient channel attention for deep convolutional neural networks. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE. <https://doi.org/10.1109/cvpr42600.2020.01155>