Neural Computing and Applications (2021) 33:11273–11284
https://doi.org/10.1007/s00521-020-05572-5

ORIGINAL ARTICLE

DENet: a deep architecture for audio surveillance applications

Antonio Greco[1] · Antonio Roberto[1] · Alessia Saggese[1] · Mario Vento[1]

Received: 11 July 2020 / Accepted: 1 December 2020 / Published online: 11 January 2021

Abstract

In the last years, a big interest of both the scientific community and the market has been devoted to the design of audio surveillance systems, able to analyse the audio stream and to identify events of interest; this is particularly true in security applications, in which the audio analytics can be profitably used as an alternative to video analytics systems, but also combined with them. Within this context, in this paper we propose a novel recurrent convolutional neural network architecture, named DENet; it is based on a new layer that we call denoising-enhancement (DE) layer, which performs denoising and enhancement of the original signal by applying an attention map on the components of the band-filtered signal. Differently from state-of-the-art methodologies, DENet takes as input the lossless raw waveform and is able to automatically learn the evolution of the frequencies-of-interest over time, by combining the proposed layer with a bidirectional gated recurrent unit. Using the feedbacks coming from classifications related to consecutive frames (i.e. that belong to the same event), the proposed method is able to drastically reduce the misclassifications. We carried out experiments on the MIVIA Audio Events and MIVIA Road Events public datasets, confirming the effectiveness of our approach with respect to other state-of-the-art methodologies.

Keywords Audio surveillance · Deep learning · Signal processing

1 Introduction

Artificial intelligence is the core component of a wide range of applications we use daily. A noteworthy example is given by surveillance applications, in which there is a strong need to automatically detect sounds of interest in a reliable way and in real time [6]. In the last years audio-only and multimodal methods [16], the latter able to combine audio and visual data, have attracted the interest of the scientific community. This branch of research aims to increase the robustness of video analytics, typically not reliable enough in presence of brightness variations, scene occlusions and other disturbances [28]. Notwithstanding, one of the main limitations of visual analysis is that it does not allow to detect some relevant events in the field of audio surveillance, such as gunshots, screams or glass breaking; in fact, they have a very distinctive audio signature that make them recognizable by analysing the audio stream.

Although it could be considered a kind of panacea for reliably managing surveillance environments, audio analysis hides several challenging pitfalls. First of all, the sounds are acquired with different types of microphones, characterized by variable cardioid diagrams, bandwidth and sensitivity; it means that the same audio track may assume very different representations depending on the audio sensor [19]. Second, the power of the audio source, its distance from the microphone and the presence of background noise, are often inconstant and highly variable characteristics in real environments that can significantly reduce the signal to noise ratio (SNR), so complicating the recognition of specific sounds [29]. Third, the duration of the events of interest is substantially variable (e.g. a gunshot is almost impulsive, while a scream or a siren is

✉ Alessia Saggese
asaggese@unisa.it

Antonio Greco
agreco@unisa.it

Antonio Roberto
aroberto@unisa.it

Mario Vento
mvento@unisa.it

[1] University of Salerno, Via Giovanni Paolo II 132, Fisciano, SA, Italy

typically longer and sustained sounds); it implies the necessity of explicitly taking into account the temporal evolution of the audio signal [12]. Finally, the amount of data available for training process in audio-related fields is rather limited, unlike the huge databases available for various computer vision tasks [13].

Therefore, an audio analysis system must be designed considering two main requirements: (i) the features and the classifiers used for the recognition of the sounds of interest must be robust with respect to the above-mentioned environmental and technological sources of noise, and, at the same time, adequately trainable even with small quantity of data; (ii) the method must include a mechanism for exploiting the temporal information of the audio signal in order to correctly classify events of different duration.

As for the first point, we can note that for several years the common practice was to search for hand-crafted representations of the audio input based on various signal processing techniques, such as the energy of the audio signal, the wavelets and the Mel-Cepstrum coefficients [11], and *classifiers* based on traditional machine learning algorithms, such as support vector machine (SVM), artificial neural network (ANN), Hidden Markov model (HMM) [5, 8, 17, 30]. In general, these kinds of methods demonstrated to be very promising, especially in presence of few data available for training; anyway, they are upper-bounded when the dataset size increases (i.e. underfitting).

The revolution introduced by trainable methods [26], and in particular by deep learning (DL) algorithms, such as deep neural network (DNN), convolutional neural network (CNN) and deep belief network (DBN) [1, 4], has changed the way for approaching this kind of problems. Indeed, these neural network architectures have been proven to be very effective when humans are unable to explain their expertise (e.g. navigation, speech recognition, and vision) [2]. The main reason which makes DL methods so powerful with respect to those based on hand-crafted features is their ability to learn a hierarchy of features directly from raw data without any prior knowledge of the specific problem at hand. These architectures create a new transformation at each level of the network as a function of the previously learned features by optimizing a problem-driven function, and therefore, do not require a target (i.e. expert knowledge). In addition, the learned features can be re-used to solve related tasks through a process called transfer learning [27]. Therefore, once fixed an effective representation learned on a similar task, it is possible to reuse it for achieving high performance also on problems that suffer from lack of reference data (e.g. rare events).

A first way for applying DL algorithms to audio problem is to create image-based representations of both the spectrum and the signal (spectrogram is probably the most popular one), with the aim to adapt well-known networks, pre-trained on computer vision tasks (e.g. ResNet, MobileNet, and so on), to the problem of audio event recognition [10, 13, 31]. Even if the improvement due to the reuse of image-based algorithms allowed to achieve state-of-the-art performance over several problems, this solution suffers of two main problems. First, the process which transforms the spectrum into an image irreversibly loses information with respect to the input signal. The impact of this loss may be small for some specific classes, while big for others. Second, even if transfer learning allows in general to reduce the amount of data required for training, the process of adapting pre-learned features to use in a different task (i.e. train all the weights of the neural network) requires anyway a quite huge amount of training data.

To address the above-mentioned problems, a common strategy is to give as input to the neural networks the raw Spectrogram and to pre-train the models over large general-purpose audio datasets [14]. Nevertheless, the Spectrogram suffers the time-frequency resolution trade-off [3]: on the one hand, a small STFT window allows to perform an analysis with high time-resolution and low-frequency resolution; on the other hand, a bigger window increases the frequency resolution at the expense of the time resolution. For this reason, the current trend is the usage of end-to-end CNNs architectures [15]: these models take as input the raw waveform and stack several strided convolutions and max-pooling layers to obtain a compact time-frequency representation of the audio signal before the classification.

In particular, SincNet [22, 23], a novel CNN architecture for speaker recognition tasks, performs well even in surveillance applications. It substitutes the first convolutional layer with a trainable band-pass filterbank. The main advantage derives from its ability to learn the frequencies-of-interest (FOI) for the events to detect and to generalize well even when dealing with small datasets, due to the reduced number of parameters to learn with respect to a more traditional convolutional layer. However, although SincNet learns the most relevant frequencies for recognizing the events of interest, it has no capability to dynamically determine the frequency components to attenuate and the ones to amplify for improving the classification in presence of overlapping background noise.

As anticipated before, another important requirement pertains the exploitation of the temporal information. Several strategies designed for taking advantage of the temporal evolution of a signal have been proposed. Starting from hand-crafted model, bag of audio features (BoF) have been widely adopted for sound event detection and classification. The model represents an audio chunk through a histogram of code-book entries which are computed by clustering low-level features over smaller frames. On the other hand, given the success achieved by neural networks in several domains, time-delay neural networks (TDNN) have been designed to deal with sequences [20]. The network uses a smaller matrix of weights (the kernel or filter), which slides over this signal and transforms it into an output using the convolution operation. This model has been generalized by 1-dimensional CNNs. More recently, recurrent neural networks (RNNs) [18] have become the cutting-edge solution for the analysis of the sequences. RNNs are a type of neural network in which the output from the previous step is fed as input to the current step. Their main limitation is due to the problem of the vanishing gradient according to the sequence size increasing. To solve this issue, gated networks, namely long-short term memory (LSTM) and gated recurrent unit (GRU), have been proposed. The gates allow the memory cells to determine when to forget certain information. Furthermore, recurrent models can be stacked after the CNN to create a convolutional recurrent neural network (CRNN). In this way, both local information and temporal context can be extracted by the model [21].

The contribution of this paper is twofold. (1) We propose a new convolutional layer, specifically designed for audio analysis problems, which has the aim of overcoming the above-mentioned issues. The proposed layer is based on the attention mechanism, that is very popular in automatic language translation and natural language processing. It basically computes a run-time weight for each part of the input sequence (i.e. the frequency components) and makes a weighted sum of each contribution. Doing that, the network is able to automatically focus the decision on a subpart of the input that is more relevant for the classification. In our case, the application of the attention mechanism allows to amplify the relevant components at the FOI (enhancement) and to attenuate the background noise (denoising). For this reason, we call it denoising-enhancement (DE) layer.

Furthermore (2), we combine the frequency-based analysis with the temporal one, by stacking on the feature extraction layers a bidirectional GRU (BGRU) cell [25] to consider the evolution of the FOI over the time in the detection process. We call the overall convolutional architecture DENet. This architecture can collect information from the past (backward) and the future (forward). This feature results in avoiding misclassifications which happen mainly with overlapping sounds and when the classification is performed at the beginning of an event-of-interest; indeed, it is hard to take a decision before analysing all its audio frames. Therefore, the analysis of the future frames introduces a paltry delay in the recognition of the event, but it is rewarded by a significant improvement in terms of audio classification accuracy. Finally, the use of a GRU instead of a LSTM for analysing the temporal evolution of the audio signal allows to use fewer parameters and, thus, to achieve a good generalization capability even in absence of a huge amount of data. As mentioned above, this is a crucial point when dealing with audio analysis problems.

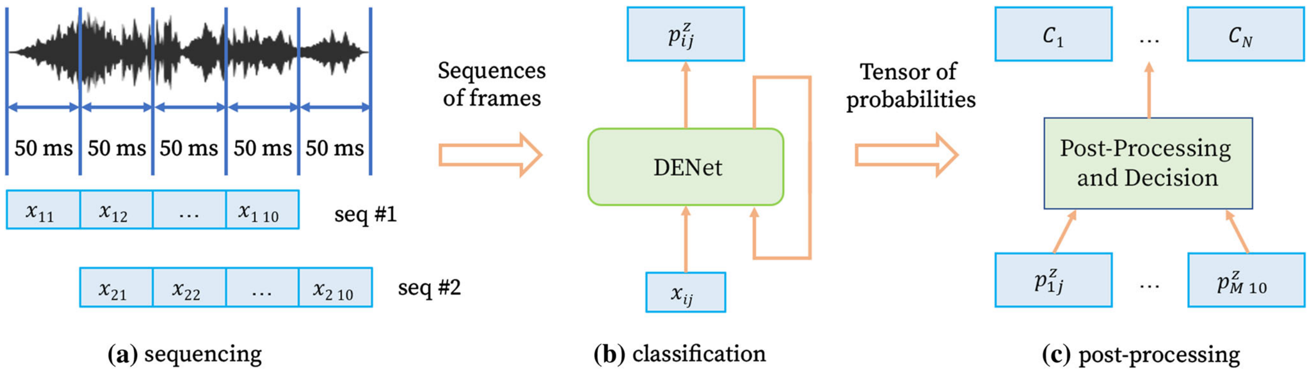
The paper is organized as follows: the proposed method is described in Sect. 2; the experimental setup, the considered datasets and the achieved results are discussed in Sect. 3. Finally, the conclusions are drawn in Sect. 4.

2 Proposed method

The architecture of the proposed system is shown in Fig. 1. The audio stream is partitioned into frames, which are in turn grouped in overlapping sequences (Fig. 1a). Hereinafter, we will refer at the j-th frame of the i-th sequence with x_{ij}. After the sequencing stage (Sect. 2.1), the frames of each sequence are classified by using a *many-to-many* bidirectional recurrent model, called denoising-enhancement network (DENet). The network takes as input directly the raw waveform, so avoiding any type of information loss. In Tables 1 and 2, we report the architectural details of the proposed network. The overall architecture, in terms of convolutional and dense layers, has been inherited by SincNet [22]. Anyway, with respect to the original architecture, (i) we performed a grid-search optimization of the layer hyperparameters over the validation set, aiming at defining the number of features maps and dense units; (ii) we introduced the novel proposed DELayer, with the aim to increase the discriminative power of the architecture; (iii) we have also added a BGRU for explicitly evaluating the temporal information.

In the first layer (Sect. 2.2.1), DENet filters the most important frequencies for the task that it is dealing with. Then, through the proposed DELayer (Sect. 2.2.2), it is able to *dynamically* identify and filter that bands which do not contain noise and, therefore, are useful for the classification, by independently analysing each one through an attention neural network. After enhancing and denoising the signal, the *time-local* feature extraction (at frame level) is performed by two additional one-dimensional convolutional layers followed by a MaxPooling.

In order to fully exploit the temporal information behind the audio stream, we stacked after the convolutional part of DENet a bidirectional gated recurrent unit (Sect. 2.2.3) which performs the extraction of the *time-global* features starting from the time-local ones of each frame in the sequence. The class probabilities of each frame in the analysed sequence are computed by a fully connected neural network. Given the i-th sequence, the output of the network consists in a matrix of probabilities, in which each



(a) sequencing **(b)** classification **(c)** post-processing

Fig. 1 System architecture. The input audio signal is divided into non-overlapped frames of size 50 ms; a sequence is composed by 10 frames (**a**). Each sequence is shifted of 1 frame w.r.t. the previous one. Then, DENet **b** computes the probabilities p^z_{ij} of each class z for each frame belonging to the analysed sequence (many-to-many classification). Finally, we apply a median decision filter **c** over the class labels of the N extracted frames; the class probability of each frame is computed by averaging all the classifications of the overlapped sequences

Table 1 DENet convolutional architecture

Layer	Info
SincNet layer	Filters: 80–Len: 251
Denoising-enhancement layer	Table 2
Max pooling	Len: 3
Convolutional layer	Filters: 80–Len: 5
Max pooling	Len: 3
Spatial dropout	Rate: 0.1
Normalization layer	
Convolutional layer	Filters: 80–Len: 5
Max pooling	Len: 3
Spatial dropout	Rate: 0.1
Normalization layer	
Bidirectional GRU layer	Units: 2048
Dropout	Rate: 0.3
Fully connected	Units: 1024
Dropout	Rate: 0.4
Fully connected	Units: 512
Dropout	Rate: 0.3
SoftMax layer	

In the *Info* column they report the parameters of each layer, i.e. the number and the length of the convolutional and pooling filters (Filters–Len), the dropout rate (Rate), and the number of units (Units) in the BGRU and fully connected layers

Table 2 DELayer—attention branch operation

Layer	Info
Convolutional layer	Filters: 30–Len: 7
Convolutional layer	Filters: 30–Len: 7
Convolutional layer	Filters: 10–Len: 7
Fully connected	Units: 128
Fully connected	Units: 64
Fully connected	Units: 1

In the *Info* column we report the parameters of each layer, i.e. the number and the length of the convolutional and pooling filters (Filters–Len) and the number of units (Units) in the fully connected layers

element p^z_{ij} corresponds to the probability that the j-th frame belongs to the class z (Fig. 1b). Finally, we predict the event label $C_{1\ldots N}$ of each of the N frames extracted from the audio stream by post-processing the probabilities resulting from the network (Fig. 1c).

Following on the previously described architecture, we can identify mainly three stages in the proposed system: *sequencing* (Sect. 2.1), *classification* (Sect. 2.2), and *post-processing* (Sect. 2.3). Each stage is extensively analysed in the following sections.

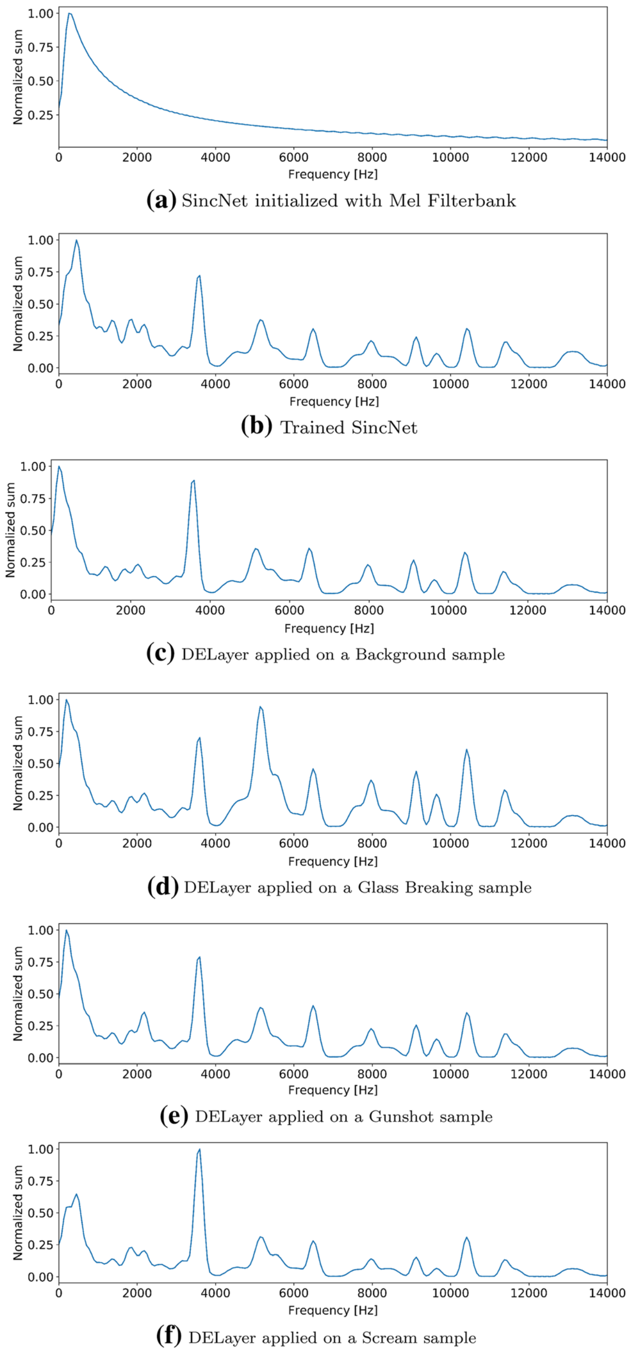
2.1 Sequencing stage

During a preliminary *sequencing* step (Fig. 1a), the audio stream acquired by the microphone is partitioned into non-overlapped frames of duration 50 ms. Then, we use the Hamming windowing function to smooth the frames in order to avoid abrupt changes at the boundaries that can cause distortions in the spectrum.

To take into account the temporal evolution of the audio signal, the input of our recurrent neural network is a sequence of 10 frames. The audio sequences are obtained by applying a sliding window of 500 ms with a shift of 50 ms, namely 1 frame. This setting has been proved in [23] to be a good choice for the considered classes of events; in this way, in fact, we are able to also detect those events localized at the boundaries.

2.2 Classification

In this subsection, we present the main building layers of the proposed DENet. For the sake of clarity, we show in Fig. 2 the reactions of the proposed layer to samples belonging to different categories of audio events. Such samples belong to the test set of the MIVIA Audio Events dataset, used in this paper for benchmarking purposes. More details about the dataset and the training procedure will be presented in Sects. 3.1 and 3.3, respectively.



(a) SincNet initialized with Mel Filterbank

(b) Trained SincNet

(c) DELayer applied on a Background sample

(d) DELayer applied on a Glass Breaking sample

(e) DELayer applied on a Gunshot sample

(f) DELayer applied on a Scream sample

Fig. 2 Comparison between the cumulative frequency responses (i.e. the normalized sum of the amplitudes of the filters) of the Mel Filterbank (**a**), the SincNet Layer (**b**) after the training phase, and the outputs of the DELayer applied on Background (**c**), Glass Breaking (**d**), Gunshot (**e**), and Scream (**f**) samples. The attention in the DELayer assigns different weights to the components at the various frequencies, in order to attenuate the noisy ones (denoising) and amplify those relevant for the classification (enhancement)

2.2.1 SincNet

Each audio sound is characterized by a spectrum which describes the distribution of the power over the frequencies. It is almost always true that the energy of a sound is located in a restricted part of the spectrum. Therefore, these frequencies that we have previously called FOI are crucial to detect and to distinguish the analysed events; on the other hand, the rest of the spectrum is a source of noise for the classification stage.

The aim of the SincNet convolutional layer, recently proposed in [22], is to extract time-local feature maps which represent the FOI. Differently from state-of-the-art CNN layers, in which all the kernel weights have to be learned, the SincNet layer is parametric. In particular, each filter consists of a sinc function in which only the cut-off frequencies are learned from the data. More formally, we can define the first convolutional layer as follows:

$$x'_{bij} = x_{ij} * g_b[f_1, f_2] \tag{1}$$

$$g_b[f_1, f_2] = 2f_2\mathrm{sinc}(f_2) - 2f_1\mathrm{sinc}(f_1) \tag{2}$$

where x'_{bij} is the filtered output, hereinafter component, g_b is the b-th pass-band function that depends from the cut-off frequencies f_1 and f_2, and $\mathrm{sinc}(f_1)$ and $\mathrm{sinc}(f_2)$ represent two sinc signals centred in f_1 and f_2, respectively.

As proposed by the authors, we initialize the cut-off frequencies of SincNet according to the Mel-scale filterbank (Fig. 2a). This scale has been chosen instead than a random initialization since it demonstrated to reproduce the nonlinear human ear perception of sound; it is more discriminative at lower frequencies and less at higher frequencies.

Starting from these weights, this layer is able to learn the FOI directly from the raw data of the class of events we are interested for our application, without the need of any intermediate frequency-based representation. In addition, the reduced number of parameters to train allows to avoid overfitting even with small datasets; this is a very important and not negligible feature, especially in case of audio analysis, since the size of the available datasets is typically quite limited.

Differently from the original layer, in the proposed model we add a L2 regularization factor to the bandwidths of the SincNet layer. In this way, we encourage the learning of filters strictly centred in the frequencies most relevant for the classification, thus reducing the possibility of overfitting. This aspect is clearly evident in Fig. 2b: the cumulative frequency response of the SincNet layer after the training phase demonstrates that, differently from the Mel initialization, there are different and strict peaks centred in the FOI.

2.2.2 Denoising-enhancement layer

SincNet allows to automatically learn the FOI, but it is not able to adapt the weights of each band-pass filter (i.e. the relevance of each frequency) to deal with the overlapping noise typically present in real environments. Starting from this consideration, we propose a new attention layer that takes as input the output of the SincNet layer and compute at runtime (i.e. for each frame) a vector of weights to assign to the components at the various frequencies. From now on we will refer to the weights vector as *attention map*.

In particular, the proposed layer dynamically determines the components affected by noise to attenuate with a small weight (denoising) and the components relevant for the classification to amplify with a bigger weight (enhancement); for this reason we named it *Denoising-Enhancement Layer*, hereinafter *DELayer*. The effect of the DELayer, applied on Background (B), Glass Breaking (GB), Gunshot (GS), and Scream (S) samples, is clearly evident in Fig. 2c, d, e, and f; we can note that the cumulative frequency response of the trained SincNet is quite different from the ones obtained after the application of the DELayer, since the learned attention maps attenuate or amplify the components at the various frequencies for each sample. Moreover, the layer has a different response for each class of the input signal.

In more detail, the output of the SincNet layer consists of several band-pass-filtered versions of the input signal. For each component, we use an attention neural network (*attention branch*) to compute a non-negative weight c_{bij}; the result for the whole spectrum is an attention map, which has a weight c_{bij} for each component. Then, we apply a SoftMax normalization over all the estimated weights; in this way, the resulting sum is equal to 1.

Finally, the output of the layer is computed as the sum of its inputs, weighted through the attention map.

Formally, the DELayer can be described as follows:

$$y_{ij} = \sum_{b=1}^{B} x'_{bij}\ w(x'_{bij}) \tag{3}$$

where y_{ij} represents the output signal, B is the number of sinc filters (i.e. the number of extracted signals) in the previous layer, and w represents the weighting function based on the attention branch output, defined as follows:

$$w(x'_{bij}) = \mathrm{softmax}(c_{bij}) \tag{4}$$

$$\mathrm{softmax}(\alpha_z) = \frac{\exp(\alpha_z)}{\sum_{\alpha_k \in \overline{\alpha}} \exp(\alpha_k)} \tag{5}$$

where α_z represents the z-th value of the attention map $\overline{\alpha}$. In this particular case, $\overline{\alpha}$ is composed by the positive coefficients computed by the attention branch for each input component x'_{bij}. For the sake of readability, the dependency between $w(x'_{bij})$ and $x'_{kij}, \forall k \neq b$ has not been reported in both the equations.

The rationale of this procedure can be traced back to the Fourier series when generalizing the sum from single frequency to frequency band signals. We are considering the input signal as the sum of the event-of-interest with an additive random noise; this assumption is reasonable in the case of audio inputs in which the background noise is overlapped with the events to detect. According to this observation, the proposed training procedure allows the DELayer to learn how to dynamically compute the Fourier coefficients necessary for reconstructing the most important frequency components of the event-of-interest.

2.2.3 Bidirectional gated recurrent unit

SincNet and DELayer allow to learn time-local features, namely these layers independently analyse each audio frame. To consider the temporal evolution of the audio signal, we stack a *Bidirectional GRU* after the convolutional layers of the proposed DENet.

The introduction of this layer is an important point in the design of our model; in fact, working at feature level, DENet is able to learn a new temporal representation which takes into account the evolution over time of the components at the FOI. In addition, the bidirectional analysis of the audio stream allows the output layer to get information from the previous and from the next states simultaneously. This is particularly important when dealing with audio events of different duration, e.g. gunshots and screams. In this way, DENet drastically reduces the misclassifications that happen at the beginning of the event-of-interest. These errors are mainly due to the partial knowledge about the temporal evolution of the events to detect. Practically, the bidirectional process acts as a sort of delayed-reasoning after the event occurrence and can improve the classification of all the frames overlapping with the target.

Furthermore, we chose a GRU instead of a LSTM in order to reduce the complexity of the proposed model. In fact, this module is characterized by less parameters and allows at the same time to reduce the number of operations and to increase the generalization capability when dealing with small datasets. These two advantages are important in our case, since we need to perform the analysis in real time and we have a reduced number of available training samples.

2.3 Post-processing stage

At this stage of our system, the output of DENet is a tensor of probabilities in which each value p^z_{ij} represents the probability that the frame j in the sequence i belongs to the class z. Due to the overlap between consecutive sequences, for each frame we have multiple different predictions. In order to exploit the redundant information, an aggregation function has been applied on all the classifications referring to the same frame to obtain a single class label. In particular, we made a class-wise average of the probabilities and chose the label with the highest probability.

At the end, to mitigate the occurrences of single-frame false positive (mainly due to very short events, like gunshots) and false negative, we adopt the median filter depicted in Fig. 3 over the decisions taken for each class (binary-class filtering). The median decision filter updates the labels by using a sliding majority rule of size 5; the value of this parameter is a good trade-off between the size of the classified frame (i.e. 50 ms) and the mean duration of the shortest event (i.e. gunshot which has an average length of 440 ms).

The use of a bidirectional layer and the mean aggregation function introduces a delay in the decision of just 1 s, which can be considered acceptable for surveillance applications.

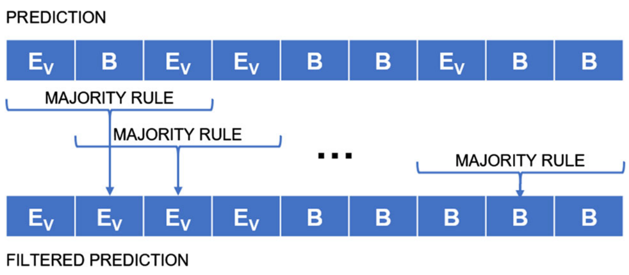


Fig. 3 Median decision filter applied to the predictions over the frames at binary-class level (E_V event, B background). It applies a sliding majority rule to remove spurious misclassifications

3 Experimental results

3.1 Datasets

We conducted our experiments to assess the performance of DENet on the MIVIA Audio Events [9] and the MIVIA Road Events [7] datasets. These databases are widely adopted by the scientific community for benchmarking purposes in the field of audio event detection in surveillance applications.

The MIVIA Audio Events dataset focuses on surveillance applications; thus, it includes glass breaking (GB), gunshot (GS), and scream (S). The class Background (B) is also included. More details on the dataset are shown in Table 3; it consists of about 30 hours of recording and includes audio files acquired with different SNRs, in the range from 5 to 30 dB; in this way, it is possible to verify the robustness of the methods with respect to environmental noise and to different distances of the source from the microphone.

Differently from the previous database, the MIVIA Road Events dataset is focused on the audio events for road surveillance applications, containing events such as tire skidding (SK) and car crash (CC). Also in this dataset, the Background (B) class is included. The audio files are divided into fourfold containing 100 events each, in order to account for cross-validation experiments. More details about the dataset are reported in Table 4.

The recordings are sampled at 32,000 Hz and quantized at 16 bits per PCM sample. Each audio file has a different background, so that several different real situations are simulated.

For both the databases, the training and the test sets have already been defined; thus, we randomly divided the training samples from the validation ones with a ratio 70–30%.

3.2 Metrics

The performance of DENet has been computed by following the experimental protocol proposed in [7].

In particular, the performance indices are useful for the final user in a real application and are, thus, *event-based*. An event is considered recognized if at least one of the frames which overlaps with the event is properly classified. On the other hand, an event is considered missed if all the frames which overlap with the event are classified as background. The remaining events are considered

Table 3 Number of events and total duration of background (B), glass breaking (GB), gunshot (G), and scream (S) recordings in the Mivia Audio Events Dataset

Type	Training set		Test set	
	Events	Duration (s)	Events	Duration (s)
B	–	58,372	–	25,037
GB	4200	6025	1800	2562
GS	4200	1884	1800	744
S	4200	5489	1800	2445

Table 4 Number of events and total duration of background (B), car crash (CC), and tire skidding (SK) recordings in the Mivia Road Events Dataset

Type	Events	Duration (s)
B	–	2732
CC	200	326
SK	200	522

classification errors. Then, these values are normalized by the number of events in the ground truth, so obtaining the recognition rate (RR), the miss rate (MR) and the error rate (ER).

As an important additional performance index, which takes into account the capability of the system to distinguish the events of interest from the background, we also compute the false-positive rate (FPR). A false positive is an event detected by the network when only a background sound is present. When a false positive is detected in two consecutive frames, we count only one error because it causes a single alarm in a real surveillance system. Finally, the FPR is computed by normalizing the number of false positives with the total number of background frames.

3.3 Training settings

The DENet architecture has been trained by using the well-known RMSprop optimizer, in which each parameter is updated separately. In particular, we have used a learning rate $lr = 10^{-4}$ and built batches of 128 samples. The hyperparameters have been tuned through a grid search on the validation set. We optimized over various architectural properties (length of the filters, number of units) and training settings (optimization algorithm, learning rate, batch size).

As for the training strategy, on the MIVIA Audio Events dataset the network is trained from scratch (TS), using the Glorot initialization procedure for convolutional and dense layers, while the Mel-filterbank bands and centre frequencies have been used for the SincNet one. The Glorot or Xavier initializer is a standard de facto for CNNs; it computes the weights with a truncated normal distribution with average 0 and standard deviation equal to $\sqrt{\frac{2}{a-b}}$, where a and b are the number of input and output units of the weight tensor. On the other hand, the chosen initialization for SincNet should amplify the frequencies of the audio signal that are suited for retaining the speech and for discarding all the other components, especially the background noise. It does not allow to start from features which perfectly represent the sounds of interest, but the convergence is surely better and easier since the weights are optimized for a similar task.

On the MIVIA Road Events dataset, in order to evaluate the possibility to transfer the learned weights between different audio event detection problems, we trained the DENet both from scratch and with a fine tuning (FT) procedure. In more detail, in the latter case we started from the weights learned by DENet on the MIVIA Audio Events dataset and fine tuned it on the training set of the MIVIA Road Events dataset. Therefore, on this dataset we can evaluate also the effect of transfer learning. This is a common choice in computer vision tasks, when no wide datasets are available for the problem of interest; indeed, most of the CNNs are trained for a variety of image classification tasks by using the original weights computed on ImageNet, in order to start from an effective representation. In spite of this, it is not a common approach in audio analysis, in which the fine tuning can have a beneficial effect due to the reduced amount of training data. By adopting this strategy, we expect an increase of the performance and a reduction of the convergence time, since the learning procedure of the network starts from a representation of the sound that is already suited for audio event detection and the data can be sufficient for an effective fine tuning.

3.4 Results

The results of the experiments performed on the MIVIA Audio Events dataset are reported in Table 5. The proposed DENet achieves a state-of-the-art recognition rate equal to 0.975, overcoming the recent method based on the trainable COPE filters applied on gammatonegrams [26] (0.960), the CNN SoundNet [4] (0.933), and two approaches based on handcrafted features [24] [7] (0.886 and 0.867). It is worth mentioning that also the standard method based on SincNet [23] is able to outperform all the other approaches (0.971). However, the proposed DELayer and the temporal information gathered with the BGRU

Table 5 Comparison of the proposed method with the state-of-the-art approaches on the MIVIA Audio Events dataset in terms of recognition rate (RR), miss rate (MR), error rate (ER), and false-positive rate (FPR)

Methods	RR	MR	ER	FPR
DENet	0.975	0.014	0.011	0.029
SincNet [23]	0.971	0.019	0.010	0.029
COPE [26]	0.960	0.031	0.009	0.043
SoundNet [4]	0.933	0.007	0.060	0.223
Haar [24]	0.886	0.099	0.014	0.014
HF + BoW + SVM [7]	0.867	0.107	0.026	0.031

The methods are ordered for descending recognition rate introduced in this paper allow to further increase the recognition rate (increase of 0.004) and to reduce the miss rate (0.014 vs 0.019), keeping at the same time the false-positive rate unchanged (0.029).

Analysing the confusion matrices of the classifications performed by SincNet and DENet on the MIVIA Audio Events dataset, reported in Table 6, we can observe that the improvement is mainly achieved on the sounds with a longer duration, namely glass breaking and scream, for which the miss rate is almost null (while SincNet has a MR of 0.044 on screams). This improvement is slightly paid on gunshot events (miss rate of 0.042), since these are sounds with reduced duration. The motivation of such experimental evidence is represented in Figs. 4 and 5. In fact, the median filter is able to reduce the false-positive rate by slightly paying in terms of recognition rate; it may thus cut out events with a short duration, especially in noisy environments.

The proposed architecture demonstrates other two important points of strength for real audio surveillance applications: the robustness to the noise and the generalization capability.

The results reported in Table 7 show that DENet is able to better preserve the performance in very noisy conditions; with SNR = 5 dB it outperforms SincNet in terms of recognition rate (0.921 vs 0.873), miss rate (0.039 vs 0.098), and false-positive rate (0.027 vs 0.029). In general, the performance of the proposed network is more stable since its standard deviation among the different SNRs is almost halved in terms of recognition rate (0.025 vs 0.044) and reduced to one-third in terms of miss rate (0.012 and 0.036). This robustness and the stability of the results as the environmental noise changes are definitely crucial capabilities in real audio surveillance applications.

As for the generalization capabilities, an experimental analysis on a different dataset was necessary. The results achieved by DENet on the MIVIA Road Events dataset are reported in Table 8, both in terms of average and standard deviation among 4 experiments, since the experimental protocol requires a fourfold cross-validation. The performance of the proposed solution on this database is even more encouraging. In fact, the network trained from scratch outperforms almost all the others in terms of recognition rate (0.975), excluding the one based on MobileNet and gammatonegrams proposed in [10] (0.995). It is noteworthy the 28.4% of performance increases with respect to the standard method based on SincNet [23] (0.975 vs 0.773), which certifies the effectiveness of the DELayer and of the BGRU for achieving better generalization capabilities. In addition, as expected, the fine tuning allows to obtain a further improvement of the recognition rate (0.998), so making the DENet trained with this procedure the state of the art even on this dataset (0.003 better than [10]). It is also important to note that DENet achieves on the MIVIA Road Events dataset a null error rate (the others have an error rate of 0.005 [10], 0.012 [26], 0.002 [5], 0.007 [8], and 0.027 [23] in the best cases); it means that each audio event detected is also properly classified.

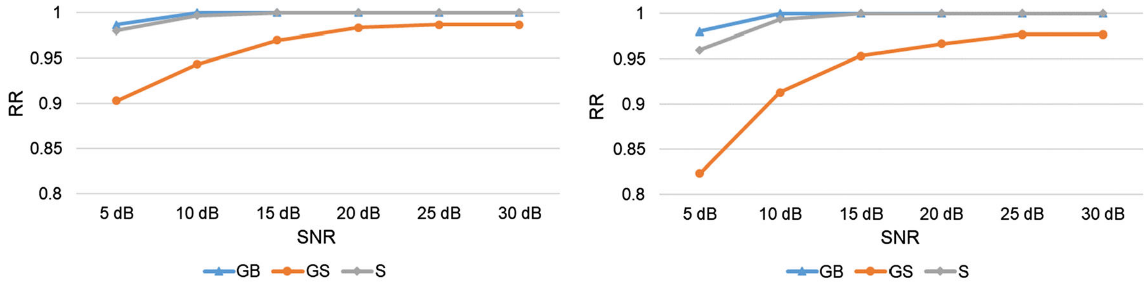
The sensitivity and the accuracy in audio event detection and classification achieved with the adoption of the DELayer and of the BGRU are partially paid in terms of FPR. Indeed, we notice an increase of this index from 0.010 to 0.043 with respect to the original network based on SincNet [23]. However, this modification not only allows to achieve a FPR comparable with that obtained by state-of-the-art methods, but implies a substantial increase in the recognition rate (0.975 vs 0.773) bringing at the same time the error rate to 0 (from 0.027). Therefore, such improvements are far superior to the slight increase of the sensitivity and, thus, justify and reward the architectural choices.

Finally, we report in Table 9 the processing time of the proposed network as a function of the buffer size, i.e. the number of sequences simultaneously classified. In particular, we evaluated the system with 1, 5, 10, and 20 sequences. We averaged the time required to predict the event probabilities over 100 evaluations for each batch size. We conducted the experiments on both CPU and GPU architectures, an Intel(R) Xeon(R) W-2133 CPU @ 3.60 GHz with 62 GB of RAM memory, and a Nvidia Titan X Pascal GPU with 12 GB of dedicated memory, respectively.

Considering a shift equal to 50 ms between two following sequences, the system must conclude the processing of the previous buffer within this time interval; however, the higher is the size of the buffer, the higher is the classification delay. We can observe that the best trade off for respecting the real-time constraint is obtained with a buffer size equal to 5 for both the GPU and CPU architectures; with this setting, a waiting time of 0.25 s, obtained

Table 6 Comparison in terms of confusion matrices and miss rate (MR) obtained by applying SincNet and DENet on the MIVIA Audio Events dataset

	SincNet				DENet			
	Detected class			MR	Detected class			MR
	GB	GS	S		GB	GS	S	
True class								
GB	0.999	0.000	0.000	0.001	0.997	0.003	0.000	0.000
GS	0.019	0.963	0.005	0.013	0.018	0.935	0.005	0.042
S	0.005	0.001	0.950	0.044	0.005	0.002	0.992	0.001



(a) Without median filter

(b) Median filter window size = 5

Fig. 4 Effect of the median filter on the recognition rate (RR) for each event class at different SNRs (from 5 to 30 dB). The filter may penalize the recognition rate of the short-duration events, such as the gunshot (GS), especially at small SNRs

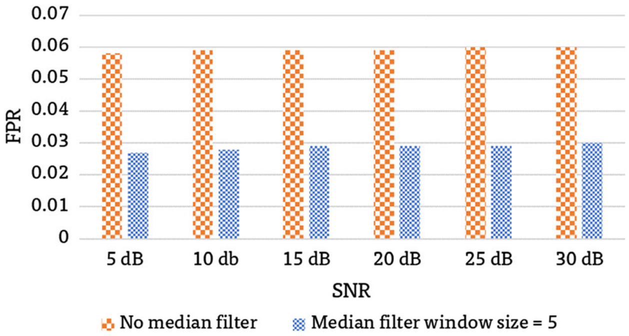


Fig. 5 Effect of the median filter on the false-positive rate (FPR) for each event class at different SNRs (from 5 to 30 dB). The filter allows to reduce the FPR of 50% in all the cases

Table 7 Results of SincNet and DENet on the MIVIA Audio Events dataset at different SNR values (from 5 to 30 dB)

SNR	SincNet			DENet		
	RR	MR	FPR	RR	MR	FPR
5 dB	0.873	0.098	0.029	0.921	0.039	0.027
10 dB	0.977	0.016	0.029	0.969	0.020	0.028
15 dB	0.992	0.002	0.029	0.984	0.011	0.029
20 dB	0.994	0.000	0.029	0.989	0.007	0.029
25 dB	0.994	0.000	0.029	0.992	0.004	0.029
30 dB	0.994	0.000	0.029	0.992	0.004	0.030
Mean	0.971	0.019	0.029	0.975	0.014	0.029
Std	0.044	0.036	0.000	0.025	0.012	0.001

In the last two rows we report the mean and the standard deviation of recognition rate (RR), miss rate (MR), and false-positive rate (FPR) computer over all the SNRs

multiplying the shift time (50 ms) with the buffer size (5), is negligible for surveillance applications.

Table 8 Comparison of the proposed method with other state-of-the-art approaches on the MIVIA Road Events dataset

Methods		RR	MR	ER	FPR
DENet (FT)	Mean	0.998	0.002	0.000	0.043
	Std	0.004	0.004	0.000	0.016
MobileNet (FT) [10]	Mean	0.995	0.000	0.005	0.037
	Std	–	–	–	–
DENet (TS)	Mean	0.975	0.025	0.000	0.021
	Std	0.026	0.026	0.000	0.010
MobileNet (TS) [10]	Mean	0.965	0.010	0.028	0.067
	Std	–	–	–	–
COPE [26]	Mean	0.940	0.048	0.012	0.040
	Std	0.043	0.049	0.013	0.018
bof [5]	Mean	0.820	0.178	0.002	0.029
	Std	0.078	0.081	1.000	0.025
bof$_{\mathrm{MFCC}}$ [8]	Mean	0.803	0.190	0.007	0.077
	Std	0.116	0.116	0.010	0.059
SincNet [23]	Mean	0.773	0.200	0.027	0.010
	Std	0.080	0.073	0.029	0.008

The methods are ordered for descending mean recognition rate (RR)

Table 9 Processing time of the proposed method on both CPU and GPU

Buffer size	CPU		GPU	
	Total (s)	Frame (s)	Total (s)	Frame (s)
1	0.1135	0.1135	0.0697	0.0697
5	0.1821	0.0364	0.0710	0.0142
10	0.2609	0.0261	0.0753	0.0075
20	0.4217	0.0211	0.0887	0.0044

The CPU is an Intel(R) Xeon(R) W-2133 CPU @ 3.60 GHz with 62 GB of RAM memory, while the GPU is a Nvidia Titan X Pascal GPU with 12 GB of dedicated memory

4 Conclusions

In this paper we propose a novel method, namely DENet, for audio event detection and classification in the field of intelligent audio surveillance.

DENet differs from the state-of-the-art methods for its capability of denoising and enhancing the input signal by combining an attention module (DELayer) with the SincNet layer; the network is thus able to attenuate the components affected by environmental noise and amplify the ones relevant for the classification.

The experimental results demonstrate that DENet is definitely more effective than the existing methods in detecting and recognizing audio events of interest on the MIVIA Audio Events and on the MIVIA Road Events benchmarks. The proposed network achieves state-of-the-art performance on both the datasets and further analyses show its robustness to noise, its stability among different environmental conditions, and its generalization capabilities.

From the results it also emerges that the proposed solution is a little bit more sensitive to false positives, as suggested by the slightly increase in terms of FPR on the MIVIA Road Events dataset. This drawback, however, can be considered negligible with respect to the improvements in terms of recognition rate, miss rate, and error rate for a security system. Anyway, a future direction for improving the proposed architecture can be the investigation of a network with two levels, the first devoted to the discrimination between background and events of interest and the second responsible for the event classification; this solution should improve the false-positive rate while retaining the high classification performance.

Funding The authors do not receive funding for this research.

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

Code availability The code is available at: https://github.com/MiviaLab/DENet.

Compliance with ethical standards

Conflict of interest The authors declare no conflict of interest.

References

1. Abdoli S, Cardinal P, Koerich AL (2019) End-to-end environmental sound classification using a 1d convolutional neural network. Expert Syst Appl 136:252–263. https://doi.org/10.1016/j.eswa.2019.06.040
2. Alom MZ, Taha TM, Yakopcic C, Westberg S, Sidike P, Nasrin MS, Esesn BCV, Awwal AAS, Asari VK (2018) The history began from alexnet: a comprehensive survey on deep learning approaches. https://arxiv.org/abs/1803.01164
3. Auger F, Flandrin P (1995) Improving the readability of time-frequency and time-scale representations by the reassignment method. IEEE Trans Signal Process 43(5):1068–1089
4. Aytar Y, Vondrick C, Torralba A (2016) Soundnet: learning sound representations from unlabeled video. In: Advances in neural information processing systems, pp 892–900
5. Carletti V, Foggia P, Percannella G, Saggese A, Strisciuglio N, Vento M (2013) Audio surveillance using a bag of aural words classifier. In: IEEE international conference on advanced video and signal based surveillance (AVSS), pp 81–86. https://doi.org/10.1109/avss.2013.6636620
6. Crocco M, Cristani M, Trucco A, Murino V (2016) Audio surveillance: a systematic review. ACM Comput Surv CSUR 48(4):1–46
7. Foggia P, Petkov N, Saggese A, Strisciuglio N, Vento M (2015) Reliable detection of audio events in highly noisy environments. Pattern Recognit Lett 65:22–28. https://doi.org/10.1016/j.patrec.2015.06.026
8. Foggia P, Petkov N, Saggese A, Strisciuglio N, Vento M (2016) Audio surveillance of roads: a system for detecting anomalous sounds. IEEE Trans Intell Transp Syst 17(1):279–288. https://doi.org/10.1109/tits.2015.2470216
9. 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. https://doi.org/10.1109/avss.2015.7301731
10. Foggia P, Saggese A, Strisciuglio N, Vento M, Vigilante V (2019) Detecting sounds of interest in roads with deep networks. In: Ricci E, Rota Bulò S, Snoek C, Lanz O, Messelodi S, Sebe N (eds) Image analysis and processing—ICIAP 2019, pp 583–592. Springer International Publishing, Cham
11. Furui S (1986) Speaker-independent isolated word recognition based on emphasized spectral dynamics. In: ICASSP'86. IEEE international conference on acoustics, speech, and signal processing, vol 11, pp 1991–1994. IEEE
12. 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
13. Greco A, Saggese A, Vento M, Vigilante V (2019) SoReNet: a novel deep network for audio surveillance applications. In: 2019 IEEE international conference on systems, man and cybernetics (SMC), pp 546–551. IEEE. https://doi.org/10.1109/smc.2019.8914435
14. 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
15. Kim T, Lee J, Nam J (2019) Comparison and analysis of sample CNN architectures for audio classification. IEEE J Sel Top Signal Process 13(2):285–297
16. Kumar P, Mittal A, Kumar P (2008) A multimodal framework using audio, visible and infrared imagery for surveillance and security applications. Int J Signal Imaging Syst Eng 1(3/4):255. https://doi.org/10.1504/ijsise.2008.026797
17. Leng YR, Tran HD, Kitaoka N, Li H (2010) Selective gammatone filterbank feature for robust sound event recognition. In: Eleventh annual conference of the international speech communication association

18. Li J, Dai W, Metze F, Qu S, Das S (2017) A comparison of deep learning methods for environmental sound detection. In: 2017 IEEE International conference on acoustics, speech and signal processing (ICASSP), pp 126–130. IEEE. https://doi.org/10.1109/icassp.2017.7952131
19. Mathur A, Isopoussu A, Kawsar F, Berthouze N, Lane ND (2019) Mic2Mic: Using cycle-consistent generative adversarial networks to overcome microphone variability in speech systems. In: Proceedings of the 18th international conference on information processing in sensor networks, pp 169–180
20. Nooralahiyan AY, Lopez L, Mckewon D, Ahmadi M (1997) Time-delay neural network for audio monitoring of road traffic and vehicle classification. In: Transportation sensors and controls: collision avoidance, traffic management, and ITS, vol 2902, pp 193–200. International Society for Optics and Photonics. https://doi.org/10.1117/12.267145
21. Purwins H, Li B, Virtanen T, Schlüter J, Chang SY, Sainath T (2019) Deep learning for audio signal processing. IEEE J Sel Top Signal Process 13(2):206–219
22. Ravanelli M, Bengio Y (2018) Speaker recognition from raw waveform with sincnet. In: 2018 IEEE spoken language technology workshop (SLT). IEEE. https://doi.org/10.1109/slt.2018.8639585
23. Roberto A, Saggese A, Vento M (2020) A deep convolutionary network for automatic detection of audio events. In: International conference on applications of intelligent systems (APPIS). https://doi.org/10.1145/3378184.3378186
24. 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. https://doi.org/10.1109/avss.2016.7738082
25. Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. IEEE Trans Signal Process 45(11):2673–2681. https://doi.org/10.1109/78.650093
26. Strisciuglio N, Vento M, Petkov N (2019) Learning representations of sound using trainable COPE feature extractors. Pattern Recognit 92:25–36. https://doi.org/10.1016/j.patcog.2019.03.016
27. Torrey L, Shavlik J (2010) Transfer learning. In: Handbook of research on machine learning applications and trends: algorithms, methods, and techniques, pp 242–264. IGI Global
28. Valera M, Velastin SA (2005) Intelligent distributed surveillance systems: a review. IEE Proc Vis Image Signal Process 152(2):192–204
29. Wan T, Zhou Y, Ma Y, Liu H (2019) Noise robust sound event detection using deep learning and audio enhancement. In: 2019 IEEE international symposium on signal processing and information technology (ISSPIT), pp 1–5. IEEE
30. Wei P, He F, Li L, Li J (2020) Research on sound classification based on SVM. Neural Comput Appl 32(6):1593–1607
31. Zhang H, McLoughlin I, Song Y (2015) Robust sound event recognition using convolutional neural networks. In: IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 559–563. https://doi.org/10.1109/icassp.2015.7178031