End-to-End Environmental Sound Classification using a 1D Convolutional Neural Network

Sajjad Abdoli[1], Patrick Cardinal, Alessandro Lameiras Koerich

Department of Software and IT Engineering, École de Technologie Supérieure, Université du Québec, H3C 1K, Montreal, QC, Canada

Abstract

In this paper, we present an end-to-end approach for environmental sound classification based on a 1D Convolution Neural Network (CNN) that learns a representation directly from the audio signal. Several convolutional layers are used to capture the signal's fine time structure and learn diverse filters that are relevant to the classification task. The proposed approach can deal with audio signals of any length as it splits the signal into overlapped frames using a sliding window. Different architectures considering several input sizes are evaluated, including the initialization of the first convolutional layer with a Gammatone filterbank that models the human auditory filter response in the cochlea. The performance of the proposed end-to-end approach in classifying environmental sounds was assessed on the UrbanSound8k dataset and the experimental results have shown that it achieves 89% of mean accuracy. Therefore, the proposed approach outperforms most of the state-of-the-art approaches that use handcrafted features or 2D representations as input. Moreover, the proposed approach outperforms all approaches that use raw audio signal as input to the classifier. Furthermore, the proposed approach has a small number of parameters compared to other architectures found in the literature, which reduces the amount of data required for training.

Keywords: Convolutional neural network, Environmental sound classification, Deep learning, Gammatone filterbank.

Abbreviations: AI: Air conditioner; BC: Between Class; CA: Car horn; CH: Children playing; CST: Chroma, Spectral contrast, Tonnetz; CNN: Convolution Neural Network; CRP: Cross Recurrence Plot; DS: Dempster–Shafer; DO: Dog bark; DR: Drilling; EN: Engine; GU: Gun shot; JA: Jackhammer; LMC: LM, CST; LM: Log-Mel; MFCC: Mel-Frequency Cepstral Coefficients; MC: MFCC, CST; SI: Siren; SKM: Spherical K-Means; ST: Street music; SVMs: Support Vector Machines.

Email addresses: `sajjad.abdoli.1@ens.etsmtl.ca` (Sajjad Abdoli), `patrick.cardinal@etsmtl.ca` (Patrick Cardinal), `alessandro.koerich@etsmtl.ca` (Alessandro Lameiras Koerich)

[1] Corresponding author

1. Introduction

In the last years, Convolutional Neural Networks (CNNs) have had significant impact on several audio and music processing tasks such as automatic music tagging (Dieleman & Schrauwen, 2014), large-scale video clip classification based on audio information (Hershey et al., 2017), music genre classification (Costa et al., 2017), speaker identification (Ravanelli & Bengio, 2018), environmental sound classification (Piczak, 2015a; Salamon & Bello, 2017; Pons & Serra, 2018; Simonyan & Zisserman, 2014; Tokozume et al., 2017; Esmaeilpour et al., 2019b), among others. Environmental sound classification is an interesting problem (Sigtia et al., 2016; Stowell et al., 2015) which has different applications ranging from crime detection (Radhakrishnan et al., 2005) to environmental context aware processing (Chu et al., 2009). Moreover, with the increasing interest in smart cities, IOT devices embedding automatic audio classification can be very useful for urban acoustic monitoring (Mydlarz et al., 2017) like intelligent audio-based surveillance system in public transportation (Laffitte et al., 2019).

Like typical automatic classification systems, most of the approaches for environmental sound classification rely on handcrafted features or learn representations from mid-level representations such as spectro-temporal features (Ludeña-Choez & Gallardo-Antolín, 2016; Costa et al., 2012). Spectral representations have been used as features in several approaches based on matrix factorization (Mesaros et al., 2015; Benetos et al., 2016; Bisot et al., 2016; Salamon & Bello, 2015; Geiger & Helwani, 2015). Mesaros et al. (2015) presented an approach for overlapping sound event detection based on learning non-negative dictionaries through joint use of spectrum and class activity annotation. Benetos et al. (2016) presented an approach for overlapping acoustic event detection based on probabilistic latent component analysis where each exemplar in a sound event dictionary consists of a succession of spectral templates. Bisot et al. (2016) learn features from time-frequency images in an unsupervised manner. The images are decomposed using matrix factorization methods to build a dictionary and the projection coefficients are used as features for classification. Salamon & Bello (2015) proposed a dictionary learning method based on the Spherical K-Means (SKM) algorithm which used log-Mel spectrograms as input. Geiger & Helwani (2015) used Gabor filterbank features and Gaussian Mixture Models for event detection. Mulimani & Koolagudi (2019) used a singular value decomposition method for extracting acoustic event specific features from spectrogram. These features are used as input to a Support Vector Machine (SVM) classifier. Recently, Xie & Zhu (2019) proposed a method for aggregation of acoustic and visual features for acoustic scene classification. Several acoustic features like spectral centroid, spectral entropy as well as several visual features like local binary pattern, histogram of gradients are proposed. A suitable feature selection algorithm like principle component analysis is also used. The selected feature set is used as input to an SVM classifier.

Recent works explore CNN-based approaches given the significant improvements over hand-crafted feature-based methods (Piczak, 2015a; Salamon & Bello, 2017; Pons & Serra, 2018; Simonyan & Zisserman, 2014; Tokozume et al., 2017). However, most of these approaches first convert the audio signal into a 2D representation (spectrogram) and use 2D CNN architectures that were originally designed for object recognition such as AlexNet and VGG (Simonyan & Zisserman, 2014; Boddapati et al., 2017). One of the main advantages of using 2D representations is that spectrograms can summarize high dimensional waveforms into a compact representation (Costa et al., 2011). Furthermore, 1D representations are noisier compared to 2D representations (Stowell & Plumbley, 2014). Piczak (Piczak, 2015a) presented a CNN with two layers followed by three dense layers. The network operates on two input channels: log-Mel spectra and their deltas. However, one of the challenges in using 2D CNNs for environmental sound classification is that the modelling capacity of such networks depends on the availability of a large amount of training data to learn kernel parameters without over-fitting. The scarcity of labeled data of environmental sounds is also a problem. Salamon & Bello (2017) presented a method based on a 2D CNN with five layers (SB-CNN) where new training samples are generated using data augmentation methods such as time stretching, pitch shifting, dynamic range compression or adding background noise (McFee et al., 2015). The 2D CNN was trained on the augmented dataset and evaluated on the original samples. They reported the classification accuracy of 79% on a dataset of environmental sounds (Salamon et al., 2014). Pons & Serra (2018) used randomly weighted 2D CNNs (non-trained) for extracting features from audio spectrograms and raw audio samples for sound classification. Several experiments have been conducted to find the best architectures for this method. In the case of environmental sound classification, the best results have been obtained by using a VGG 2D CNN (Simonyan & Zisserman, 2014) as a feature extractor and SVMs as classifiers. They reported mean accuracy of 70% for this problem. Boddapati et al. (2017) used spectrogram, Mel-Frequency Cepstral Coefficients (MFCC) and Cross Recurrence Plot (CRP) and AlexNet and GoogLeNet for classification of 2D representations of environmental sounds. They reported accuracy between 92% and 93% on classifying environmental sounds. Tokozume et al. (2017) proposed a new method called Between-Class (BC) learning for training neural networks. The network, for which the input is a mixture of two audio samples, is trained to predict the mixing ratio of the samples. According to their experiments, the BC learning has shown performance improvement for various architectures used for sound identification tasks. They also proposed an end-to-end 1D CNN (EnvNet-v2) that performs well on various environmental sound datasets when trained with the BC learning approach, compared to conventional learning techniques. The best error rate of 8.6% is reported on ESC-10 dataset (Piczak, 2015b).

1D CNNs that learn acoustic models directly from audio waveforms are becoming a popular method in audio processing due to the ability of these networks to take advantage of the signal's fine time structure (Hoshen et al., 2015). Kim et al. (2018) proposed a 1D CNN architecture for music auto-tagging inspired by the building blocks of Resnets (He et al., 2016) and SENets (Hu et al., 2018). Zhu et al. (2016) proposed an end-to-end learning approach for speech recognition based on multiscale convolutions that learns the representation directly from audio waveforms. Three 1D convolutional layers with different kernel sizes are used for feature extraction and the features are concatenated by a pooling layer for ensuring a consistent sampling frequency for the rest of the network. They reported 23.28% of word error rate on a dataset drawn from a collection of sources including read, conversational, accented, and noisy speech. Ravanelli and Bengio (Ravanelli & Bengio, 2018) proposed the SincNet, an end-to-end approach for speaker identification and verification. The first layer of such a model is based on parametric sinc functions, which are band-pass filters. Only low and high cutoff frequencies of the filters are learned from data. This model learns meaningful filters for the first layer and decreases the number of parameters of the model. This model achieves a sentence error rate of 0.85% on TIMIT dataset (Garofolo et al., 1993). Zeghidour et al. (2018) also proposed an end-to-end 1D CNN architecture for speech recognition by learning a filter bank which is considered as a replacement of Mel-filterbanks. Hoshen et al. (2015) proposed an end-to-end multichannel 1D CNN for speech recognition. They also found that the timing difference between channels is an indicator of the location of the input in space. They reported 27.1% of single channel word error rate on a large vocabulary voice search dataset. Sainath et al. (2015) used a similar architecture for speech recognition. They showed that features learned directly from the audio waveform match the performance of log-Mel filterbank energies. Dai et al. (2017) proposed several very deep convolutional models for environmental sound classification that achieved 72% of accuracy on UrbanSound8k dataset. The proposed models consist of batch normalization, residual learning, and down-sampling in the initial layers of the CNN.

The prediction of two CNNs that learn from raw audio and 2D representations of the signal can also be combined to achieve a robust prediction. Li et al. (2018) combined one network that learns directly from audio waveform (RawNet) and one network that learns high level representations from log-Mel features (MelNet). The models are trained independently and the prediction of the two models is combined using the Dempster–Shafer (DS) method. This ensemble method produces 92.2%, 92.6% and 83.1% of accuracy on UrbanSound8k (Salamon et al., 2014), ESC-10 and ESC-50 (Piczak, 2015b) datasets, respectively. Su et al. (2019) proposed the TSCNN-DS model, which also combined the prediction of two CNNs using the DS method. Five auditory features such as Log-Mel spectrogram (LM), MFCC, Chroma, Spectral contrast and Tonnetz (CST) are extracted from the audio signal. LM and CST are stacked and considered as one feature set (LMC). Likewise, MFCC and CST (MC) features are also combined by stacking. The two feature sets are used for training two identical four-layer CNNs. The prediction of the CNNs are then combined using the DS method. The TSCNN-DS achieves the classification accuracy of 97.2% on UrbanSound8k dataset.

In this paper, we propose an end-to-end 1D CNN for environmental sound classification that learns the representation directly from the audio signal instead of from 2D representations (Piczak, 2015a; Salamon & Bello, 2017, 2015). The proposed end-to-end approach provides a compact architecture that reduces the computation cost and the amount of data required for training. With the aim of extracting relevant information directly from audio waveforms, several convolutional layers are used to learn low-level and high-level representations. The highest level of representation is then used for classifying the input signal by means of three fully connected layers. Experimental results on UrbanSound8k dataset, which contains 8,732 environmental sounds from 10 classes, have shown that the proposed approach outperforms other approaches based on 2D representations such as spectrograms (Piczak, 2015a; Salamon & Bello, 2017; Pons & Serra, 2018; Salamon & Bello, 2015) by between 11.24% (SB-CNN) and 27.14% (VGG) in terms of mean accuracy. Furthermore, the proposed approach does not require data augmentation or any signal pre-processing for extracting features.

Our contribution in this paper is twofold. We present an end-to-end 1D CNN initialized with Gammatone filterbanks that has few parameters and which does not require a large amount of data for training compared to dense 2D CNNs which have millions of trainable parameters. Besides, it achieves state-of-the-art performance. Secondly, the proposed approach can handle audio signals of any length by using a sliding window of appropriate width that breaks up the audio signal into short frames of dimension compatible with the input layer of the end-to-end 1D CNN.

This paper is organized as follows. Section 2 presents the ideas behind the proposed end-to-end 1D CNN architecture and the proposed approach to deal with variable audio lengths. We also present the variations in the architecture that may arise from different input dimensions as well the process of aggregating the predictions on audio frames. Section 3 presents the benchmarking dataset, the experimental protocol, the procedure used to fine-tune the proposed 1D CNN to the data, the evaluation of different input sizes, the enhancements in the proposed 1D CNN to improve its performance and an analysis of the frequency response of the filters learned at the different convolutional layers. In Section 4, we compare the performance of the proposed approach with the state-of-the-art in environmental sound classification and we analyze the magnitude responses of the filters learned at the first convolutional layer to gain some insight on the behaviour of the proposed 1D CNN. Finally, the conclusions and perspectives of future work are presented in the last section.

2. Proposed End-to-End Architecture

The aim of the proposed end-to-end architecture is to handle audio signals of variable lengths, learning directly from the audio signal, a discriminative representation that achieves a good classification performance on different environmental sounds.

2.1. Variable Audio Length

One of the challenges of using 1D CNNs in audio processing is that the length of the input sample must be fixed but the sound captured from the environment may have various duration. Therefore, it is necessary to adapt a CNN to be used with audio signals of different lengths. Moreover, a CNN must be used for continuous prediction of input audio signals of environmental sounds.

One way to circumvent this constraint imposed by the CNN input layer is to split the audio signal into several frames of fixed length using a sliding window of appropriate width. Therefore, in our approach we use a window of variable width to conditionate the audio signal to the input layer of the proposed 1D CNN. The window width depends mainly on the signal sampling rate. Furthermore, successive audio frames may also have a certain percentage of overlapping, which aim is to maximize the use of information. This naturally increases the number of samples as some parts of the audio signal are reused and that can be viewed as some sort of data augmentation. The process of framing the audio signal into appropriate frames is illustrated in Figure 1.

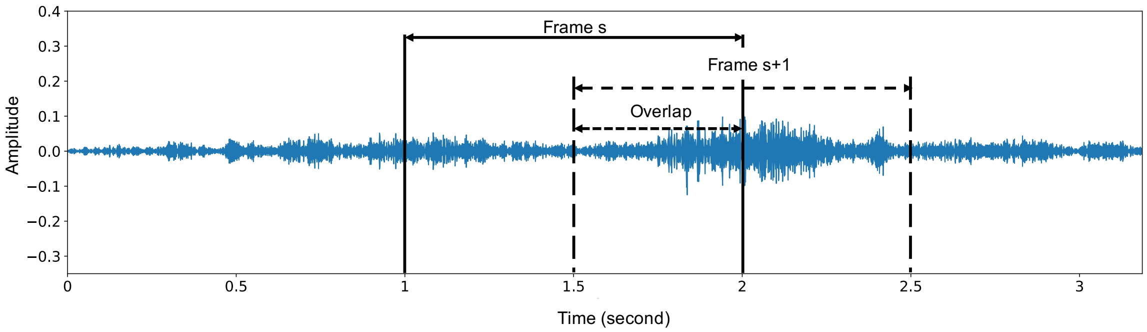


Figure 1: Framing the input audio signal into several frames $(s, s+1)$ with appropriate overlapping percentage (50%).

Moreover, the sampling rate of the audio signals has a direct impact on the dimensionality of the input sample and eventually on the computational cost of model. For environmental sounds, a sampling rate of 16 kHz may be considered a good trade-off between the quality of the input sample and the computational cost of the model.

2.2. 1D CNN Topology

A 1D CNN is analogous to a regular neural network but but it has generally raw data as input instead of handcrafted features. Such an input data is processed through several trainable convolutional layers for learning an appropriate representation of the input. According to the *"local connectivity"* theorem, the neurons in a layer are connected only to a small region of the previous layer. This small region of connectivity is called a receptive field. The input to out 1D CNN is an array representing the audio waveform, which is denoted as X. The network is designed to learn a set of parameters Θ to map the input to the prediction T according to a hierarchical feature extraction given by Equation 1:

$$T = F(X \mid \Theta) = f_L(...f_2(f_1(X \mid \Theta_1) \mid \Theta_2) \mid \Theta_L) \tag{1}$$

where L is the number of hidden layers in the network. For the convolutional layers, the operation of the l-th layer can be expressed as:

$$T_l = f_l(X_l \mid \Theta_l) = h(W \otimes X_l + b), \quad \Theta_l = [W, b] \tag{2}$$

where $\otimes$ denotes the convolution operation, X_l is a two-dimensional input matrix of N feature maps, W is a set of N one dimensional kernels (receptive field) used for extracting a new set of features from the input array, b is the bias vector, and $h(\cdot)$ is the activation function. The shapes of X_l, W and T_l are (N, d), (N, m) and $(N, d-m+1)$, respectively. Several pooling layers are also applied between the convolutional layers for increasing the area covered by the next receptive fields. The output of the final convolutional layer is then flattened and used as input of several stacked fully connected layers, which can be described as:

$$T_l = f_l(X_l \mid \Theta_l) = h(WX_l + b), \quad \Theta_l = [W, b] \tag{3}$$

In the case of multiclass classification, the number of neurons of the output layer is the number of classes. Using softmax as the activation function for the output layer, each output neuron indicates the membership degree of the input samples for each class. During the training process, the parameters of the network are adjusted according to the back-propagated classification error and the parameters of the network are optimized to minimize an appropriate loss function (Goodfellow et al., 2016).

The proposed topology aims a compact 1D CNN architecture with a reduced number of parameters. The number of parameters of a CNN is directly related to the computational effort to train such a network as well as to the need of a large amount of data for training. Therefore, the proposed architecture shown in Figure 2 is made of four convolutional layers, possibly interlaced with max pooling layers, followed by two fully connected layers and an output layer. The baseline model shown in Figure 2 has as input an array of 16,000 dimensions, which represents 1-second of audio sampled at 16 kHz. However, this is not a constraint since we can adapt the model for different audio lengths and sampling rates in two ways: (i) change the model architecture to adapt it to the characteristics of the audio inputs; (ii) padding or segmenting the audio piece to adapt it to the input dimensions of the network.

Several other configurations can also be derived from subtle modifications of the base model (shown in Figure 2) to adapt it to shorter or longer audio inputs, as shown in Table 1. This implies modifying the number of convolutional layers as well as the number and the dimension of filters and the stride. However, for long contiguous audio recordings, instead of increasing the input dimension of the network, which also implies increasing the number of parameters, and consequently its complexity, it is preferable to split the audio waveform into shorter frames by changing the window width as explained in Section 2.1. In this way, we keep the network compact and it can process audio waveforms of any length. In spite of that, in Section 3.2 we evaluate different audio lengths as input, keeping a fixed sampling rate of 16 kHz.

The proposed 1D CNN has large receptive fields in the first convolutional layers since it is assumed that the first layer should have a more global view of the audio signal. Moreover, the environmental sound signal is non-stationary *i.e.* the frequency or spectral contents of the signal changes with respect to time. Therefore, shorter filters do not provide a general view on the spectral contents of the signal. The output of

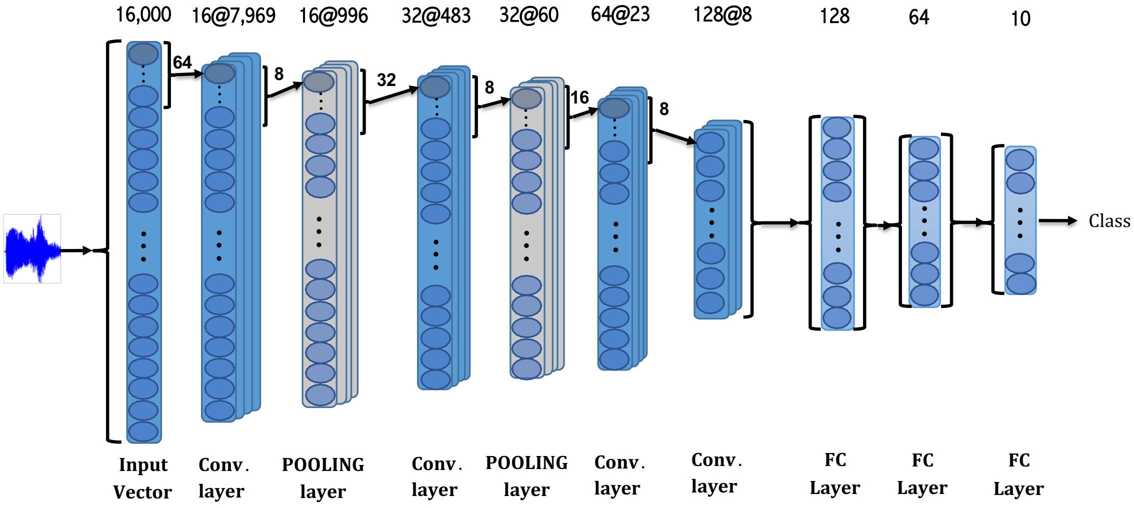


Figure 2: The architecture of the proposed end-to-end 1D CNN for environmental sound classification. The dimension, number of filters and filter size are given for the input size of 16,000. For other input sizes, the values are presented in Table 1.

Table 1: The configuration of the convolutional layers (CL) and pooling layers (PL) for the end-to-end CNN considering different input sizes (audio lengths).

Input Size		Layer							
		CL1	PL1	CL2	PL2	CL3	CL4	CL5	PL3
50,999	Dim	25,468	3,183	1,576	197	91	42	20	5
	# Filters	16	16	32	32	64	128	256	256
	Filter Size	64	8	32	8	16	8	4	4
	Stride	2	8	2	8	2	2	2	4
32,000	Dim	15,969	1,996	983	122	54	24	11	2
	# Filters	16	16	32	32	64	128	256	256
	Filter Size	64	8	32	8	16	8	4	4
	Stride	2	8	2	8	2	2	2	4
16,000	Dim	7,969	996	483	60	23	8	NA	NA
	# Filters	16	16	32	32	64	128	NA	NA
	Filter Size	64	8	32	8	16	8	NA	NA
	Stride	2	8	2	8	2	2	NA	NA
16,000G	Dim	15,489	19,36	953	119	52	23	NA	NA
	# Filters	64	64	32	32	64	128	NA	NA
	Filter Size	512	8	32	8	16	8	NA	NA
	Stride	1	8	2	8	2	2	NA	NA
8,000	Dim	3,969	496	233	29	7	NA	NA	NA
	# Filters	16	16	32	32	64	NA	NA	NA
	Filter Size	64	8	32	8	16	NA	NA	NA
	Stride	2	8	2	8	2	NA	NA	NA
1,600	Dim	785	392	189	94	44	NA	NA	NA
	# Filters	16	16	32	32	64	NA	NA	NA
	Filter Size	32	2	16	2	8	NA	NA	NA
	Stride	2	2	2	2	2	NA	NA	NA

NA: Not Applicable. G: First layer of the CNN initialized with Gammatone filterbank.

the last pooling layer for all feature maps is flattened and used as input to a fully connected layer. In order to reduce the over-fitting, batch normalization is applied after the activation function of each convolution layer (Ioffe & Szegedy, 2015). The last fully connected layer has ten neurons. Mean squared logarithmic error, defined in Equation 4 is used as loss function ($\mathcal{L}$):

$$\mathcal{L} = \frac{1}{N}\sum_{i}^{N} log(\frac{p_i + 1}{a_i + 1})^2 \tag{4}$$

where p_i, a_i and N are the predicted class, the actual class, and the number of samples respectively.

For all input sizes shown in Table 1, after the last pooling layer, there are two fully connected layers with 128 and 64 neurons respectively on which a drop-out is applied with a probability of 0.25 for both layers (Srivastava et al., 2014). The ReLU activation function ($h(x) = max(x, 0)$) is used for all layers, except for the output layer where a softmax activation function is used. Since the amount of data for training is limited, it is not feasible to use deeper architectures without significant over-fitting. By the use of the architecture shown in Figure 2, it is possible to omit a signal processing module because the network is powerful enough to extract relevant low-level and high-level information from the audio waveform.

The convolutional layers of the proposed architecture are inspired in Aytar et al. (2016) who proposed a CNN architecture (SoundNet) for learning sound representations from unlabeled videos. The SoundNet (Aytar et al., 2016) learns multimodal features from audio and video using two concurrent CNNs which are further used with a SVM classifier. On the other hand, the proposed 1D CNN architecture learns the representation directly from the audio waveform, and it uses such a learned representation as input to a fully connected neural network for classification.

2.3. Gammatone Filterbanks

Another interesting characteristic of such a 1D CNN is that its first layer can be initialized as a Gammatone filter bank. A Gammatone filter is a linear filter described by an impulse response of a gamma distribution and a sinusoidal tone. This initialization can be viewed as a trade-off between handcrafted features and representation learning. In this configuration, the kernels of the first layer are initialized by 64 band-pass Gammatone filters with central frequency ranging from 100 Hz to 8 kHz. Such a filterbank decomposes the input signal into 64 frequency bands.

Gammatone filters have been used in models of the human auditory system and are physiologically motivated to simulate the structure of peripheral auditory processing stage. For this reason, Gammatone filters have also been used to initialize the first layer of 1D CNNs for automatic speech recognition (Hoshen et al., 2015; Zeghidour et al., 2018; Sainath et al., 2015). Figure 3 illustrates the frequency response of the Gammatone filterbank, generated by the Gammatone-like spectrograms toolbox developed by Ellis (2009).

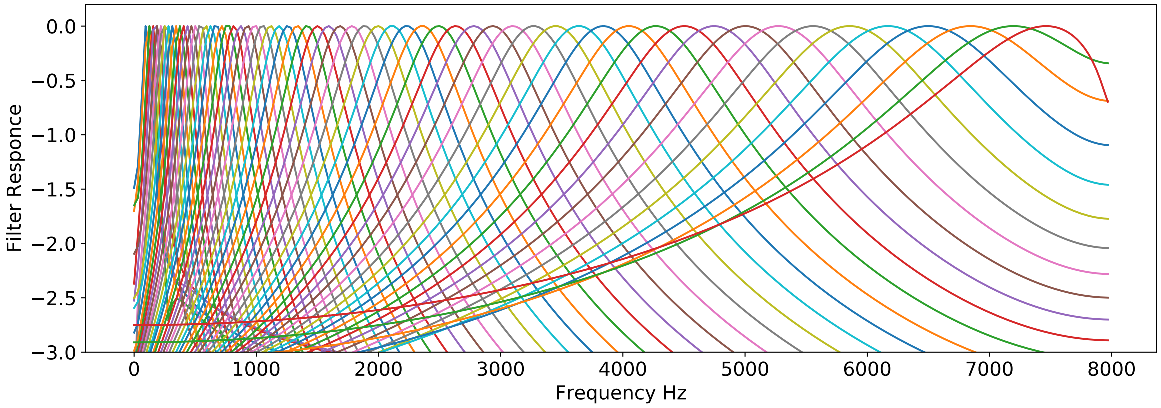


Figure 3: Frequency response of 64 filters of Gammatone filterbank.

2.4. Aggregation of Audio Frames

In the case where the input audio waveform X is split into S frames denoted as $X_1, X_2, \ldots, X_S$, during the classification we need to aggregate the CNN predictions to come up to a decision on X, as illustrated in Figure 4. For such an aim, different fusion rules can be used to reach a final decision, such as the majority vote or the *sum* rule, which are denoted in Equations 5 and 6 respectively.

$$y_i = \sum_{j=1}^{S} o_{ji} \tag{5}$$

where o is the CNN prediction for the $j = 1, \ldots, S$ segment of the audio waveform X and $i = 1, \ldots, K$ is the predicted class. S is the number of frames and K is the number of classes.

$$y_i = \frac{1}{S} \sum_{j=1}^{S} o_{ji} \tag{6}$$

When there are K classes, we generate K values and them for an audio input, we choose the class with the maximum y_i value:

$$\text{Choose} \quad C_i \quad \text{if} \quad y_i = \max_{k=1}^{K} y_k \tag{7}$$

3. Experimental Results

The proposed end-to-end 1D CNN for environmental sound classification was evaluated on the UrbanSound8k dataset (Salamon et al., 2014). This dataset consists of 8,732 audio clips summing up to 7.3 hours of audio recordings. The maximum duration of audio clips is four seconds. The classes and the number of samples in each class are: ”Air conditioner (AI): 1000”, ”Car horn (CA): 429”, ”Children playing (CH): 1000”,

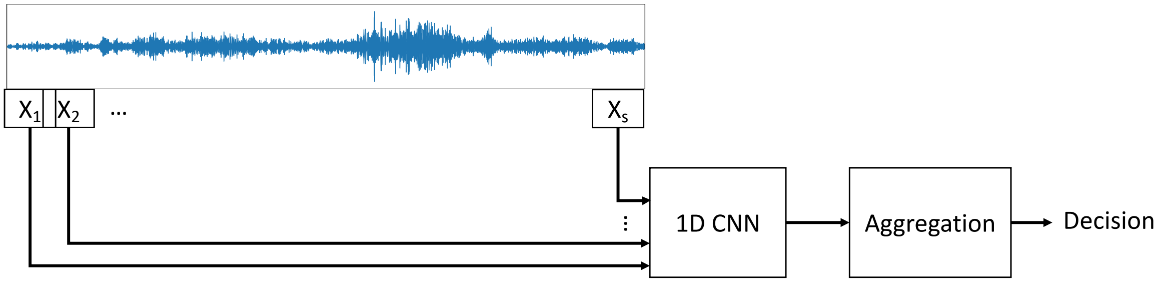


Figure 4: Aggregation of the predictions on the audio frames.

”Dog bark (DO): 1000”, ”Drilling (DR): 1000”, ”Engine (EN) idling: 1000”, ”Gun shot (GU): 374”, ”Jackhammer (JA): 1000”, ”Siren (SI): 929”, ”Street music (ST): 1000”. The original audio clips are recorded at different sample rates. For the experiments presented in this paper, they have been downsampled to 16 kHz in order to unify the shape of the input signal for the 1D CNN.

3.1. Fine-Tuning the 1D CNN Architecture

The number of convolutional layers plays a key role in detecting high-level concepts. The number of convolutional layers for the base model shown in Figure 2 was determined in an exploratory experiment using the audio files of the UrbanSound8k dataset. The audio files were segmented into 16,000 samples and successive frames have 50% of overlapping. Ten percent of the dataset was used as validation set and 10% percent of the dataset was also used as test set. Each network was trained with 80% of the dataset up to 100 epochs with batch sizes of 100 samples. The accuracy achieved by the 1D CNN with one to four convolutional layers on test set was 69%, 75%, 79% and 80%, respectively. Four convolutional layers is the upper limit since the minimal dimension of the feature map has been reached at this layer. The same procedure was also adopted to find the best number of convolution layers as well as their parameters for the other configurations derived from the base model which are described in Table 1.

3.2. Evaluation on Different Audio Lengths

All experiments reported in this subsection used a 10-fold cross-validation procedure to produce a fair comparison with the results reported by Salamon et al. (2014). One of the nine training folds is used as validation set for optimizing the parameters of the network to achieve the best accuracy. A batch size of 100 samples was used for training the CNNs and they were trained up to 100 epochs with early stopping. The Adadelta (Zeiler, 2012) optimizer with the default learning rate of 1.0 was used. Adadelta has been chosen because this method dynamically adapts the learning rate during the optimization process.

First, the proposed end-to-end 1D CNN is evaluated on different audio lengths to assess the impact of the input length on the classification performance. Next, the full audio recordings of UrbanSound8k dataset, which have 50,999 frames ($\approx$ three seconds), were also segmented into shorter frames using a sliding window and considering different overlapping percentages (0%, 25%, 50%, and 75%). The architecture shown in Figure 2 was adapted according to the parameters described in Table 1, leading to audio frames of 1,600 ($\approx$ 100 msec), 8,000 ($\approx$ 500 msec), 16,000 ($\approx$ 1 second) and 32,000 samples ($\approx$ 2 seconds).

The process of segmenting the audio signal into frames and aggregating the predictions of the classifier for all frames, resembles the process of aggregating the prediction of ensemble of classifiers. In this process, the most important parts of the audio signal contribute more to the final decision while the noisy or outlier frames have their importance averaged during the aggregation process. Table 2 shows the best results achieved with different frame sizes, window overlapping and combination rules on the UrbanSound8k dataset in terms of mean accuracy. For the classification of each test sample of the original dataset, the predictions for each audio segment are combined using either the majority voting or the sum rule (Kittler et al., 1998). Table 2 shows that the 16,000-input architecture achieved the highest accuracy which is the same accuracy achieved by 1D CNN with 50,999 inputs, even if it has almost twice less parameters than that network. Furthermore, the 8,000-input architecture achieved a mean accuracy close to that, even if it has almost three times less parameters. If we increase the input size from one second to two or more seconds, besides increasing the number of parameters of the models, we reduce the number of audio segments, which may affect the training of such models due to the reduced amount of data. For this reason, we do not observe any improvement for audio segments beyond one second (16,000 frames). On the other hand, for the 1,600-input architecture, the mean accuracy is about 6% lower than the best architectures. This is an indication that short audio segments do not contain enough information to train properly the 1D CNN. However, this behaviour may be particular for the UrbanSound8k dataset and it cannot be generalized to other audio classification tasks or datasets. Table 2 also shows the computational time per epoch for training the networks with a subset 10,000 audio segments. The input size has also a direct relationship with training time, as more operations need to be done for larger inputs. Therefore, the 16,000-input CNN provides the best tradeoff between the number of parameters of network, computational time and mean accuracy.

Table 2: Mean accuracy and standard deviation on the UrbanSound8k dataset over the 10 folds for the different architectures (input dimensions) and 50% overlapping.

Input Dimension	**Combination Rule**	**Mean±SD Accuracy**	**# of Parameters**	**Computational Time (Sec)**
50,999	NA	**83%**±1.3%	421,146	3.917
32,000	Maj Voting	82%±0.9%	322,842	3.325
16,000	Sum Rule	**83%**±1.3%	256,538	1.863
8,000	Sum Rule	80%±1.9%	116,890	1.073
1,600	Sum Rule	77%±3.0%	394,906	0.648

NA: Not applicable

The box-plot of Figure 5 also shows that the 16,000-input 1D CNN is the best choice since it provides the highest median; the interquartile range is the smallest one; and there is no outlier. Furthermore, such an architecture has the same mean accuracy, but almost half of the number of parameters than the second-best choice, the 50,999-input 1D CNN. Therefore, the 16,000-input 1D CNN is preferable over other architectures, as it presents the best trade-off between the number of parameters and accuracy.

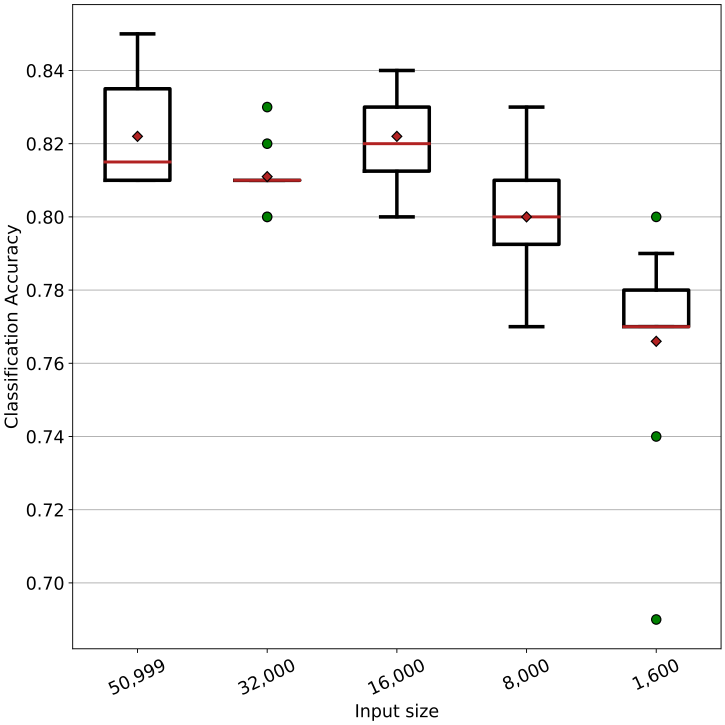


Figure 5: The box plot for the five different input sizes on UrbanSound8k dataset.

In order to have a better insight about the behavior of the convolutional filters learned by the proposed 1D CNN, the Fourier transform of some filters was computed and their frequency responses are shown in Figure 6. These filters were randomly initialized and trained for the specific task and all of their parameters, such as central frequency, bandwidth, gain/attenuation, were learned directly from the data with the aim of minimizing a loss function. The learned filters are a combination of different (mainly band-pass and band-reject) filters with selective attenuation levels for different frequency levels. The filters of the first layers (CL1 and CL2) do not exhibit dominant frequencies and are quite noisy. On the other hand, the filters learned at the deeper layers (CL3 and CL4) are more regular filters, i.e., they have a well-defined frequency response which is closer to ideal filters. However, the resolution of the Fourier transform of the deeper layers is lower than in the initial layers because they are smaller than the initial ones. This analysis lead us to propose some enhancements to the proposed approach as an attempt to improve the response of the filters learned by the network.

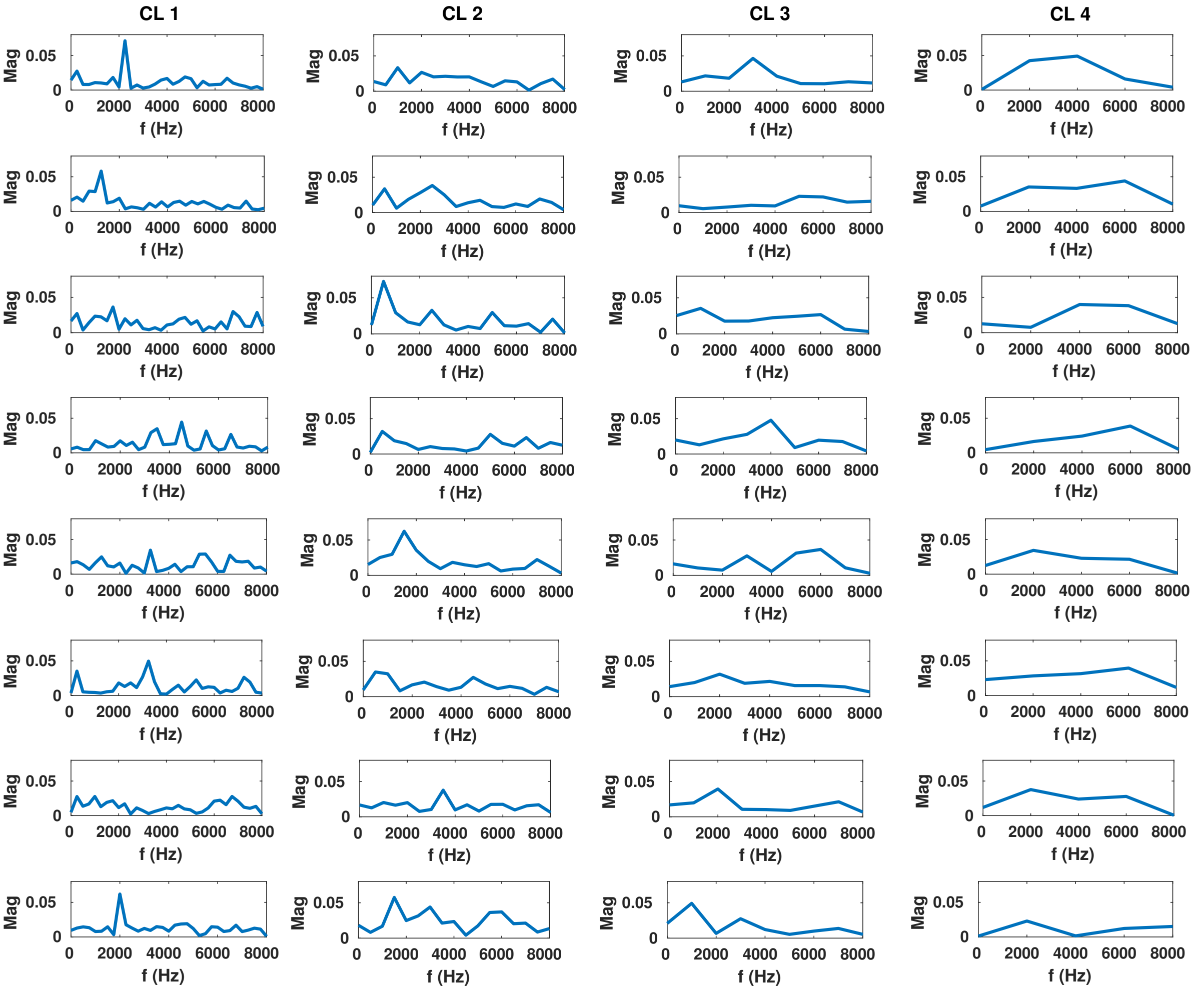


Figure 6: Fourier transform of randomly selected filters from the four convolutional layers (CLs) of the proposed 16,000-input 1D CNN shown in Figure 2.

3.3. Architecture Enhancement

Three enhancements to the proposed approach are evaluated: (i) replacing the Hamming sliding window by a rectangular window because the Hamming window smooths the signal and reduces the energy of the beginning and end of the audio frame and this may cause a loss of information; (ii) augmenting slightly the amount of training data by increasing the window overlapping during the audio segmentation; (iii) initializing the first convolutional layer as a Gammatone filterbank as described in Section 2, and make this layer non-trainable. During the training procedure, these filters can be modified by the forward and backward propagation. However, the best performance is achieved by making these filters non-trainable.

Table 3 summarizes the three proposed enhancements and their impact on the mean accuracy and on the computational time per epoch during training. The rectangular window leads to a slight improvement of 2% in the mean accuracy. Increasing the overlapping from 50% to 75% led to another 2% of improvement in the mean accuracy. Finally, initializing the first layer of such a 1D CNN with a Gammatone filterbank, also contributed to improve the mean accuracy in 2%, even if the number of parameters doubles due to the increase in the number of filters in such a layer. This also increases the training time. An important remark is that all these enhancements have also improved the performance of most of the other 1D CNN architectures presented in Table 1. In spite of that, the 16,000-input 1D CNN remains the one with the highest mean accuracy.

Table 3: Improvements in the mean accuracy for the 16,000-input 1D CNN on the UrbanSound8k dataset.

CL1 Initialization	**Window**	**Overlapping**	**Combination Rule**	**Mean Accuracy**	**# of Parameters**	**Computation Time (Sec)**
Randomly	Hamming	50%	Sum Rule	83%	256,538	1.863
Randomly	Rectangular	50%	Sum Rule	85%	256,538	1.863
Gammatone	Rectangular	50%	Sum Rule	87%	550,506	6.099
Randomly	Rectangular	75%	Sum Rule	87%	256,538	1.863
Gammatone	Rectangular	75%	Sum Rule	**89%**	550,506	6.099

Figure 7 shows the Fourier transform of some of the filters of the enhanced model with non-trainable Gammatone filterbank. Similar to the filters of the original model (Figure 6), the filters of the deepest layers (CL3 and CL4) have a well-defined frequency response. Filters of the intermediate layer (CL2) still do not exhibit dominant frequency levels. Even thought, the minor changes in the responses of the intermediate and deeper filters, the Gammatone filters of the first layer were useful to improve the mean accuracy of the proposed 1D CNN.

4. Discussion

Table 4 shows the mean classification accuracy achieved by the proposed 1D CNN as well as the results achieved by other state-of-the-art approaches described in the literature. The proposed 1D CNN achieved a mean accuracy of 89% with a standard

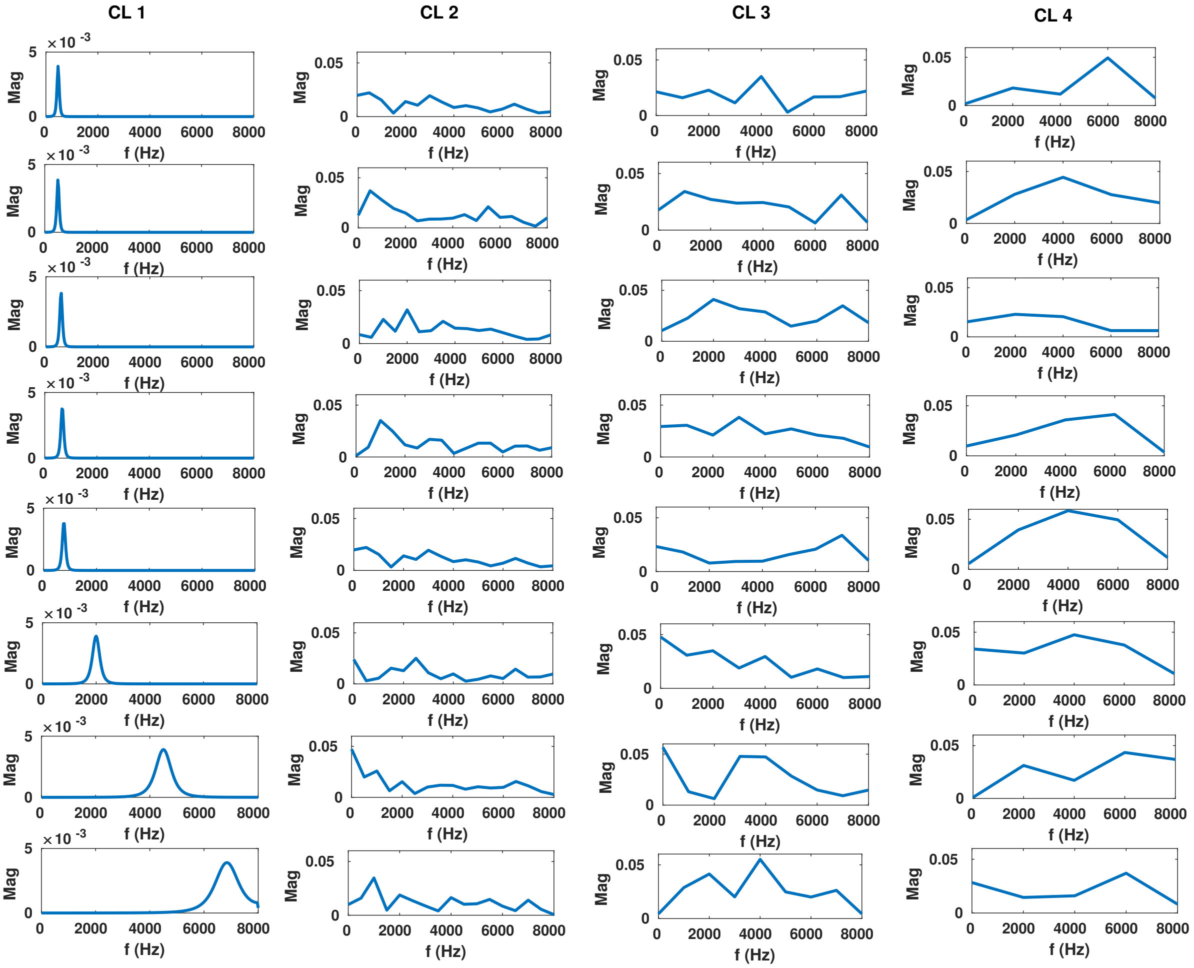


Figure 7: Fourier transform of randomly selected filters from the four convolutional layers (CLs) of the 16,000-input 1D CNN with Gammatone filterbank in the first convolutional layer of the network.

deviation of only 0.9% across the 10 folds. The proposed 1D CNN, the RawNet (Li et al., 2018), the EnvNet-v2 (Tokozume et al., 2017) and the M18 CNN (Dai et al., 2017) are end-to-end architectures, which learn the representation directly from the audio waveform. DS-CNN is a combinational model, which uses both raw audio signal and 2D representations as input to a CNN. All other approaches in Table 4 use 2D representations of the audio signal as input. As it is shown in Table 4, the proposed approach has lower number of parameters than most of the state-of-the-art approaches described in the literature and therefore it requires a relatively few number of samples for appropriate training. Furthermore, it is shown that the proposed algorithm outperforms all other approaches which use raw audio signal as input to the CNN. Therefore, the proposed approach is a quite suitable candidate to be used in ensemble models as described by Li et al. (2018). Figure 8 also compares the proposed 1D CNN with other approaches on UrbanSound8k dataset for environmental sound classification using a boxplot generated from the accuracy scores of 10 folds. Note that for some models the information about the accuracy scores of 10 folds was not available. So, only mean accuracy of the models are reported.

The proposed approach also does not require any signal processing module for feature extraction from audio signal. Therefore, it is suitable to be used in mobile or embedded devices. Moreover, as mentioned by Boddapati et al. (2017), the operation of generating 2D representations from audio signal is time-consuming. For instance, producing spectrograms of ESC-50 (Piczak, 2015b) dataset which consists of 2,000 samples takes five minutes. Generating corresponding MFCC features also takes five minutes. In addition to that, producing CRP representations takes 24 hours. Also, 2D representations can not be computed on GPU due to lack of suitable libraries. This issue makes models based on 2D representations impractical for real-time applications.

Moreover, approaches based on 2D representations are much more vulnerable to adversarial attacks which can easily fool these models. As it is shown by Esmaeilpour et al. (2019a), the models based on 2D representations can be easily fooled by adversarial attacks originally designed to fool image processing models. They have also pointed out that generalizing the current attacks to raw audio signals is not feasible because of the high-dimensionality of raw audio signals.

Figure 9 shows the confusion matrix of the proposed end-to-end 1D CNN on the UrbanSound8k dataset. Values along the diagonal indicate the number of samples classified correctly for each specific class. It shows that the ST and CH classes are the hardest classes for the CNN. However, EN and GU classes are well separated by the proposed CNN.

4.1. Filter Response

The magnitude responses of the convolutional filters of the first layer of the proposed 1D CNN are shown in Figure 10. To obtain a better image representation of the frequency response, the number of kernels in the first layer has been increased to 64 (compared to 16 in the one used in the experiments). Note that this configuration led to a slight decrease in the classification accuracy. Figures 10(a) and 10(b) show the

Table 4: Mean accuracy of different approaches on the UrbanSound8k dataset.

Approach	**Representation**	**Mean**	**# of**
		Accuracy	**Parameters**
TSCNN-DS (Su et al., 2019)	2D	97%	15.9 M
GoogLeNet (Boddapati et al., 2017)	2D	93%	6.7 M
MelNet (Li et al., 2018)	2D	90%	211 k
SB-CNN (DA) (Salamon & Bello, 2017)	2D	79%	241 k
SKM (DA) (Salamon & Bello, 2015)	2D	76%	NA
SKM (Salamon & Bello, 2015)	2D	74%	NA
PiczakCNN (Piczak, 2015a)	2D	73%	26 M
SB-CNN (Salamon & Bello, 2017)	2D	73%	241 k
VGG (Pons & Serra, 2018)	2D	70%	77 M
DS-CNN (Li et al., 2018)	1D-2D	92%	NA
Proposed 1D CNN Gamma	**1D**	**89%**	**550 k**
RawNet (Li et al., 2018)	1D	87%	377 k
Proposed 1D CNN Rand	1D	87%	256 k
EnvNet-v2 (Tokozume et al., 2017)	1D	78%	101 M
M18 CNN (Dai et al., 2017)	1D	72%	3.7 M

NA: Not available. DA: With data augmentation.

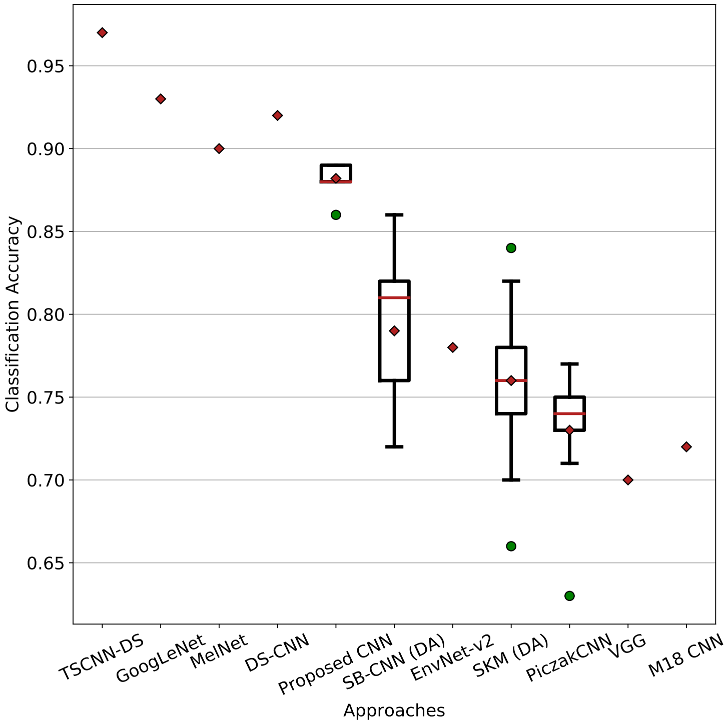


Figure 8: Classification accuracy of the proposed 1D CNN as well as the results obtained by other state-of-the-art approaches. Some parts of figure adapted from (Salamon & Bello, 2017).

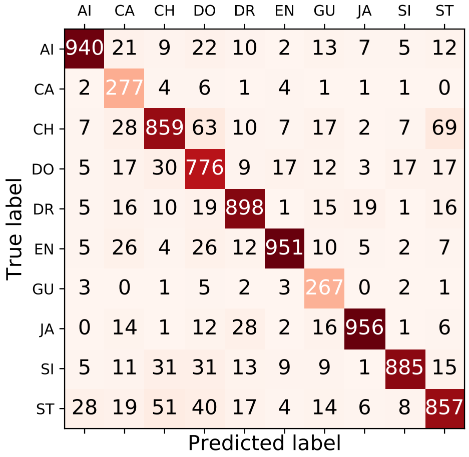
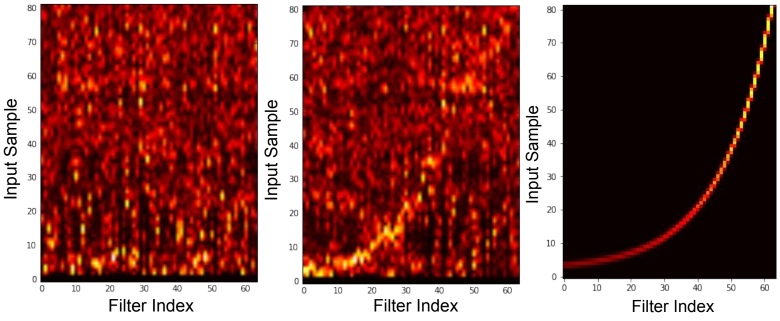


Figure 9: Confusion matrix for the proposed end-to-end 1D CNN.

response of the filters after convergence and the response of the kernels sorted based on their central frequencies, respectively. The central frequency of each kernel is computed by computing the Fast Fourier transform of the filter and by selecting the frequency bin with the highest peak. Each row in the image is created by feeding the network with a sinusoidal wave with a specific frequency. For such an aim, sinusoidal waves in the range of 1 Hz to 8 kHz, with a step of 100 Hz, have been used. The feature map of the first convolutional layer is first obtained and then, it is computed the average of the feature map along the time axis. Figure 10(c) shows the output of 64 Gammatone filters used as band-pass filters.



(a) (b) (c)

Figure 10: Magnitude response of the convolutional filters of the first layer of the end-to-end 1D CNN: (a) response of the filters after convergence; (b) response of the kernels after sorting based on their central frequency; (c) frequency response of band pass filters. Center frequency of filters are selected according to constant Q transform rules (Brown, 1991).

From Figure 10(b), it can be seen that the learned filters have a logarithmic response similar to the band pass filters created using cardinal sinusoidal functions. In addition, this behavior is also similar to how humans perceive sounds, which is also logarithmic (Roederer, 2008). A similar behavior has also been observed in other end-to-end systems for audio processing tasks (Hoshen et al., 2015; Sainath et al., 2015; Tokozume & Harada, 2017).

5. Conclusion

In this paper, an end-to-end 1D CNN for environmental sound classification has been proposed. The architecture of the network consists of three to five convolutional layers, depending on the length of the audio signal. The proposed end-to-end CNN learns the representation directly from the audio signal. The proposed approach was evaluated on a dataset of 8,732 audio samples and the experimental results have shown that the proposed end-to-end approach learns several relevant filter representations which allows it to outperform most of state-of-the-art approaches based on 2D representations and 2D CNNs. It also performs better than all models that use raw audio signal as input and use UrbanSound8k dataset (Salamon et al., 2014) for environmental sound classification. Furthermore, the proposed end-to-end 1D architecture has fewer parameters than most of the other CNN architectures for environmental sound classification. Moreover, the proposed approach does not require any signal processing module for audio classification, which makes this model quite suitable to be used in mobile sound recognition applications or in embedded systems.

However, even if we have achieved the best results using 1D representation of the audio signal, it may have a complementarity between the learned 1D filters and the filters learned from 2D representations (spectrograms), at least for some classes. This is an indication that the overall performance may be improved by combining the approaches that use 1D and 2D representations similar to the ensemble model proposed by Li et al. (2018). As a future work, we will investigate if such a combination is feasible and if it can lead to a better performance in classifying environmental sounds. Furthermore, the filters learned in the intermediate convolutional layers of the proposed 1D CNN do not exhibit dominant frequencies and seems to be noisy. A further investigation is necessary to find out how to circumvent such a problem and possibly improve further the performance of the proposed 1D CNN.

Availability of Data and Material

UrbanSound8k dataset (Salamon et al., 2014) is used for training and testing the method. The dataset is available online [2]. and the source code of the proposed end-to-end CNN will also be made available in the final version of the paper.

[2] https://urbansounddataset.weebly.com/urbansound8k.html

Competing Interests

The authors declare that they have no competing interests.

Credit Authorship Contribution Statement

Sajjad Abdoli: Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Data Curation, Writing – Original Draft, Visualization. **Patrick Cardinal and Alessandro Lameiras Koerich**: Conceptualization, Methodology, Validation, Formal Analysis, Investigation, Resources, Data Curation, Writing – Review & Editing, Supervision, Project Administration, Funding Acquisition.

Acknowledgements

This work was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC). This work was also supported by the NVIDIA GPU Grant Program.

References

Aytar, Y., Vondrick, C., & Torralba, A. (2016). Soundnet: Learning sound representations from unlabeled video. In *Advances in neural information processing systems* (pp. 892–900).

Benetos, E., Lafay, G., Lagrange, M., & Plumbley, M. (2016). Detection of overlapping acoustic events using a temporally-constrained probabilistic model. In *IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 6450–6454).

Bisot, V., Serizel, R., Essid, S., & Richard, G. (2016). Acoustic scene classification with matrix factorization for unsupervised feature learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 6445–6449).

Boddapati, V., Petef, A., Rasmusson, J., & Lundberg, L. (2017). Classifying environmental sounds using image recognition networks. *Procedia computer science*, *112*, 2048–2056.

Brown, J. C. (1991). Calculation of a constant Q spectral transform. *The Journal of the Acoustical Society of America*, *89*, 425–434.

Chu, S., Narayanan, S., & Kuo, C. (2009). Environmental sound recognition with time–frequency audio features. *IEEE Transactions on Audio, Speech, and Language Processing*, *17*, 1142–1158.

Costa, Y., Oliveira, L., Koerich, A., Gouyon, F., & Martins, J. (2012). Music genre classification using LBP textural features. *Signal Processing*, *92*, 2723–2737.

Costa, Y. M., Oliveira, L. S., & Silla, C. N. (2017). An evaluation of Convolutional Neural Networks for music classification using spectrograms. *Applied Soft Computing*, *52*, 28–38. doi:10.1016/j.asoc.2016.12.024.

Costa, Y. M. G., Oliveira, L. E. S., Koerich, A. L., & Gouyon, F. (2011). Music genre recognition using spectrograms. In *18th International Conference on Systems, Signals and Image Processing* (pp. 1–4).

Dai, W., Dai, C., Qu, S., Li, J., & Das, S. (2017). Very Deep Convolutional Neural Networks for Raw Waveforms. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 421–425).

Dieleman, S., & Schrauwen, B. (2014). End-to-end learning for music audio. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 6964–6968).

Ellis, D. P. W. (2009). *Gammatone-like spectrograms, web resource.*. URL: http://www.ee.columbia.edu/~dpwe/resources/matlab/gammatonegram/.

Esmaeilpour, M., Cardinal, P., & Koerich, A. L. (2019a). A robust approach for securing audio classification against adversarial attacks. *arXiv preprint arXiv:1904.10990*, (pp. 1–14).

Esmaeilpour, M., Cardinal, P., & Koerich, A. L. (2019b). Unsupervised feature learning for environmental sound classification using cycle consistent generative adversarial network. *arXiv preprint arXiv:1904.04221*, (pp. 1–34).

Garofolo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., & Pallett, D. S. (1993). Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. *NASA STI/Recon technical report n*, *93*.

Geiger, J., & Helwani, K. (2015). Improving event detection for audio surveillance using Gabor filterbank features. In *23rd European Signal Processing Conference* (pp. 714–718).

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep learning. MIT Press Cambridge volume 1.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Hershey, S., Chaudhuri, S., Ellis, D., Gemmeke, J., Jansen, A., Moore, R., Plakal, M., Platt, D., Saurous, R., Seybold, B. et al. (2017). Cnn architectures for large-scale audio classification. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 131–135).

Hoshen, Y., Weiss, R. J., & Wilson, K. W. (2015). Speech acoustic modeling from raw multichannel waveforms. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4624–4628).

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).

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning* (pp. 448–456).

Kim, T., Lee, J., & Nam, J. (2018). Sample-level cnn architectures for music auto-tagging using raw waveforms. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 366–370).

Kittler, J., Hatef, M., Duin, R. P., & Matas, J. (1998). On combining classifiers. *IEEE transactions on pattern analysis and machine intelligence*, *20*, 226–239.

Laffitte, P., Wang, Y., Sodoyer, D., & Girin, L. (2019). Assessing the performances of different neural network architectures for the detection of screams and shouts in public transportation. *Expert Systems with Applications*, *117*, 29–41.

Li, S., Yao, Y., Hu, J., Liu, G., Yao, X., & Hu, J. (2018). An ensemble stacked convolutional neural network model for environmental event sound recognition. *Applied Sciences*, *8*, 1152.

Ludeña-Choez, J., & Gallardo-Antolín, A. (2016). Acoustic event classification using spectral band selection and non-negative matrix factorization-based features. *Expert Systems with Applications*, *46*, 77–86.

McFee, B., Humphrey, E., & Bello, J. (2015). A software framework for musical data augmentation. In *International Society for Music Information Retrieval Conference* (pp. 248–254).

Mesaros, A., Heittola, T., Dikmen, O., & Virtanen, T. (2015). Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 151–155).

Mulimani, M., & Koolagudi, S. G. (2019). Segmentation and characterization of acoustic event spectrograms using singular value decomposition. *Expert Systems with Applications*, *120*, 413–425.

Mydlarz, C., Salamon, J., & Bello, J. (2017). The implementation of low-cost urban acoustic monitoring devices. *Applied Acoustics*, *117*, 207–218.

Piczak, K. (2015a). Environmental sound classification with convolutional neural networks. In *25th International Workshop on Machine Learning for Signal Processing* (pp. 1–6).

Piczak, K. J. (2015b). Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia* (pp. 1015–1018). ACM.

Pons, J., & Serra, X. (2018). Randomly weighted cnns for (music) audio classification. *arXiv preprint*, . URL: `https://arxiv.org/pdf/1805.00237.pdf`.

Radhakrishnan, R., Divakaran, A., & Smaragdis, A. (2005). Audio analysis for surveillance applications. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (pp. 158–161).

Ravanelli, M., & Bengio, Y. (2018). Speaker recognition from raw waveform with sincnet. In *2018 IEEE Spoken Language Technology Workshop (SLT)* (pp. 1021–1028). IEEE.

Roederer, J. G. (2008). The physics and psychophysics of music: an introduction. Springer Science & Business Media.

Sainath, T. N., Weiss, R. J., Senior, A., Wilson, K. W., & Vinyals, O. (2015). Learning the speech front-end with raw waveform cldnns. In *Sixteenth Annual Conference of the International Speech Communication Association* (pp. 1–5).

Salamon, J., & Bello, J. (2015). Unsupervised feature learning for urban sound classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 171–175).

Salamon, J., & Bello, J. (2017). Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, *24*, 279–283.

Salamon, J., Jacoby, C., & Bello, J. (2014). A dataset and taxonomy for urban sound research. In *22nd ACM International Conference on Multimedia* (pp. 1041–1044). New York, NY, USA.

Sigtia, S., Stark, A., Krstulović, S., & Plumbley, M. (2016). Automatic environmental sound recognition: Performance versus computational cost. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *24*, 2096–2107.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint*, . URL: `https://arxiv.org/pdf/1409.1556.pdf`.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.

Stowell, D., Giannoulis, D., Benetos, E., Lagrange, M., & Plumbley, M. (2015). Detection and classification of acoustic scenes and events. *IEEE Transactions on Multimedia*, *17*, 1733–1746.

Stowell, D., & Plumbley, M. D. (2014). Automatic large-scale classification of bird sounds is strongly improved by unsupervised feature learning. *PeerJ*, *2*, e488. doi:https://doi.org/10.7717/peerj.488.

Su, Y., Zhang, K., Wang, J., & Madani, K. (2019). Environment sound classification using a two-stream cnn based on decision-level fusion. *Sensors*, *19*, 1733.

Tokozume, Y., & Harada, T. (2017). Learning environmental sounds with end-to-end convolutional neural network. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 2721–2725).

Tokozume, Y., Ushiku, Y., & Harada, T. (2017). Learning from between-class examples for deep sound recognition. *arXiv preprint*, . URL: https://arxiv.org/pdf/1711.10282.pdf.

Xie, J., & Zhu, M. (2019). Investigation of acoustic and visual features for acoustic scene classification. *Expert Systems with Applications*, *126*, 20–29.

Zeghidour, N., Usunier, N., Synnaeve, G., Collobert, R., & Dupoux, E. (2018). End-to-end speech recognition from the raw waveform. *arXiv preprint*, . URL: https://arxiv.org/pdf/1806.07098.pdf.

Zeiler, M. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint*, . URL: https://arxiv.org/pdf/1212.5701.pdf.

Zhu, Z., Enge, J. H., & Hannun, A. (2016). Learning multiscale features directly from waveforms. *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, (pp. 1305–1309).