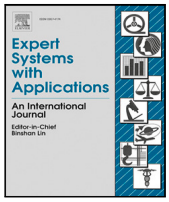
Expert Systems With Applications 249 (2024) 123768

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



A Lightweight Channel and Time Attention Enhanced 1D CNN Model for Environmental Sound Classification

Huaxing Xu, Yunzhi Tian, Haichuan Ren, Xudong Liu *

School of Electrical and Information Engineering, Zhengzhou University, Zhengzhou, Henan, 450001, China

ARTICLE INFO

Keywords:
Environmental sound classification
1D CNN
Attention
Snapshot ensemble

ABSTRACT

One dimension convolutional neural networks (1D CNN) that directly take raw waveforms as input has less competition than 2D CNN recognizing environmental sound. In order to overcome its disadvantages, we propose a novel lightweight 1D CNN structure by employing attention mechanism, which has significant improvement in both accuracy and computational complexity. Concretely, (1) two attention modules are constructed along channel and time dimension separately, and combined to give an intermediate feature map, which focus on key frequency band and semantically related time frame information. (2) Without increasing training overhead, snapshot ensemble is employed to further improve performance. Results from two benchmarking datasets (UrbanSound8k, ESC-10) demonstrated that: by employing attention mechanism, our model outperforms all of the previously reported 1D CNN approaches in accuracy with less parameters. Meanwhile with improved performance gain, the proposed model is superior than most of the existing spectral-based 2D CNN approaches and competitive with SOTA performance, while with orders of magnitude parameters fewer. Overall, it indicates our model is compact and has good potential in practical resource-limited applications, such as sound recognition on embedded platform.

1. Introduction

In recent years, environmental sound classification (ESC) that aims to classify sound recordings based on their characteristics has been developed rapidly in the research field of audio signal processing (Chu et al., 2009; Virtanen et al., 2018). ESC can be easily implemented in many application in practice, such as automatic surveillance (Ntalampiras et al., 2009), infants monitoring, elderly care, etc. However, identifying environmental sounds is generally difficult due to the intricate nature of peripheral environment, the environment complexity such as diversity of sound production, overlapping of multiple sources, and the absence of high-level structure (Aucouturier et al., 2007; Phan et al., 2016). To complicate matters further, environment is filled with background noises in realistic, those ambient noises blur the boundary between the target and other environmental sounds, which result in performance degradation.

Since ESC is developed from acoustic signal processing, early it mainly deals with feature extraction, feature analysis, and classifier design. Traditional techniques in extracting audio features include Mel Frequency Cepstral Coefficients (MFCC) (Eronen et al., 2005), Gammatone features (Valero & Alias, 2012), Log-Mel Spectrograms and Wavelet features (Valero & Alias, 2012). Other signal processing technique like dictionary learning (Chu et al., 2009) and Matrix Factorization (MF) (Bisot et al., 2017) are also employed as feature analysis methods. Classic classifiers including support vector machines (SVM), Gaussian mixture model (GMM) (Vuegen et al., 2013), hidden Markov models (HMMs), K-nearest neighbor (KNN) (Dhanalakshmi et al., 2009) has also been studied thoroughly (reader may refer to Chandrakala and Jayalakshmi (2019), Crocco et al. (2016), Singh et al. (2023) for an in depth overview). Although traditional algorithms may obtain reasonable accuracy, deficits truly exist, especially for large datasets.

- First, feature extraction, feature analysis and classifier design are usually constructed separately, which in practice requires tedious feature-to-classifier design process.
- Second, above mentioned classifiers (Dhanalakshmi et al., 2009; Vuegen et al., 2013) are unable to learn deep information about audio signals effectively due to its inherent shallow structures, especially when ESC target is interfered by environmental noises.

Recently, deep learning models, particularly the convolutional neural networks (CNNs), have exhibited outstanding performance and gradually becoming dominant approach in acoustic signal processing (Singh et al., 2023). In literatures, pre-extracting features before sending to the CNNs are very commonly seen. To make CNN model

* Corresponding author.
E-mail addresses: xuhuaxing@zzu.edu.cn (H. Xu), 289432092@qq.com (Y. Tian), haichuan.ren@zzu.edu.cn (H. Ren), xudongl@zzu.edu.cn (X. Liu).

https://doi.org/10.1016/j.eswa.2024.123768
Received 5 July 2023; Received in revised form 4 January 2024; Accepted 18 March 2024
Available online 21 March 2024

process one-dimensional audio sequences normally as image process does, different types of spectrograms extraction such as MFCC (Nguyen & Pernkopf, 2018), mel-based nearest neighbor filter (NNF) spectrogram (Nguyen & Pernkopf, 2018), constant-Q transform (CQT) (Zeinali et al., 2018), gammatonegram (Phan et al., 2017), and scattering spectra (Zhang et al., 2019) were adopted. With those pre-extracted spectrograms as front-end, various CNN models including vanilla VGG, ResNet and improved transformer networks can be chosen as back-end. Some methods known as transfer learning (Kong et al., 2020; Kumar et al., 2018) and knowledge distillation (Kumar & Ithapu, 2020; Tripathi & Paul, 2022) can also be employed to extract more robust and transferable features. Although some excellent results can be achieved with these model frameworks, there still lies some limitations:

- Firstly, feature extraction module concatenated with CNN/Transformer classifier structure is not a complete end-to-end process. With image-like spectrogram dominates pre-extracting features, it requires a convention from 1D audio signal into 2D images at first, which involves parameter selection like frame length, window size, hop length, number of Mel-filters, et cetera. For example, in Huzaifah (2017), authors found the optimal window size during transformation depends on the characteristics of the audio signal itself. In some extent, these hand-crafted feature framework may not suitable for general purpose ESC task.
- Secondly, the optimal selection of audio feature type varies from different ESC task. For instance, Zhang et al. (2018) compared the performance between the log-mel and loggammatone features, reported that log-gammatone alone performed better. However (Pham et al., 2021) demonstrated that combining multi-types of features for some ESC task can improve performance further since different spectral features can provide complementary information.
- Lastly, spectrogram-related features tend to form a deeper network. Usually deeper networks mean stronger representational ability, but also easier to get overfit due to its large number of parameters (Purwins et al., 2019; Sigtia et al., 2016), especially cases where the training data is not enough. Meanwhile, deeper network needs more memorize size, computation resource and inference time. For the applications where lightweight or real-time required, such as on embedded systems, computing-constrained IoT devices (Purwins et al., 2019), these issues present huge obstacles to ESC deployment.

To address current limitations, one dimension CNN (1D CNN) without pre-extracting features has become another study trend recently (Abdoli et al., 2019; Dai et al., 2017; Gupta et al., 2022). It extracts feature information directly on the one dimension raw audio samples. For example, in Aytar et al. (2016), authors trained 1D CNN with a huge unlabeled video dataset, the whole process can be viewed as knowledge transferring from 2D image to 1D sound. In Tokozume and Harada (2017), an end-to-end 1D CNN ESC is proposed, it can achieve higher performance over log-mel 2D CNN. Recently, by introducing Gammatone filters initializing the kernels of the first layer to improve model performance, Abdoli et al. (2019) proposed a small size 1D CNN architecture with only five convolution layers for ESC. Compared with 2D CNN-based ESC, 1D CNN-based ESC has several advantages:

- Without feature pre-extraction and additional knowledge of signal pre-processing, 1D CNN model can exploit the hidden information of signal structure and not worry about the information loss caused by 1D-to −2D conversion. Consequently, it forms a more concise end-to-end framework. On the other hand, by using the sliding window method, 1D CNN model is compatible with any arbitrary length audio sequence, which also grant it potential in processing any general-purpose audio-related classification task.
- Instead of 2D matrices, 1D arrays are used during training and inference phase for both kernels calculation and feature mapping in CNN layers, the computational complexity and size of model can be significantly reduced. Besides, only involving frame splitting further avoiding computationally intensive front-end spectrum calculation. In this way, the application of real-time can be easily implemented on the resource-constrained devices.

Despite advantages it has, the performance test on datasets UrbanSound8k and ESC-10 among the literature shows that the highest accuracy achieved by the existing models of 1D CNN is still surpassed by the highest accuracy 2D CNN models. To the best of our knowledge, for commonly-used UrbanSound8k database, the top accuracy of single 1D CNN model (Abdoli et al., 2019) is about 89%, but many 2D CNN models proposed in İnik (2023), Mushtaq and Su (2020), Song et al. (2022) have exceeded 95%. By investigation of literatures, we argue that the notable performance gap can attributed to the following reasons:

First, the central building block of 1D CNN models in Abdoli et al. (2019), Aytar et al. (2016), Dai et al. (2017), Tokozume and Harada (2017), Tokozume et al. (2018) mainly relies on different convolutions and pooling, or extra modules such as Residual block, batch normalization (Dai et al., 2017) to further improve performance. However, environmental sound in real is intermittent and always interfered by clutter of multiple noises (Abdoli et al., 2019), that is to say not all the sound frame contains valued information. For example, in public ESC datasets, there are many periods of silence in recorded audio clips, only a few of the frames are actually useful. Taking into account of many irrelevant sound frames will further degrade the performance of model. Secondly, it has been widely accepted in acoustic signal processing area that the frequency band has more characteristics information about target sound than the time series. In other word, different frequency band information makes contributions for sound recognition in 2D CNN models trained with spectrogram, but this kind of rich relation information among diverse frequencies cannot be well exploited by 1D CNN models. Thirdly, although the degree varies, limited receptive field issue is more severe in 1D convolution. It is due to the fact that, even with the same ratio of kernel size, 1D convolution kernel search sound target related information only along the time axis, which contains less characteristics information than the spectrogram domain has. On the other hand, thanks to highlight of characteristics information, compared with 1D CNN, 2D CNN model could utilize the attention mechanism more efficiently to locate the related sound frames or frequency bands, which helps to avoid salient features that are less relevant to the target sound and makes it easier to improve performance (Wu & Zhang, 2021; Zhang et al., 2021).

Motivated by the success of attention mechanisms (Hu et al., 2018; Wang et al., 2019) and goal to address the aforementioned performance gap, in this paper, we proposed an enhanced 1D CNN model with two attention modules for general-purpose ESC task. Without incurring additional training costs, snapshot ensemble training methods was also incorporated to further enhance the performance of this model. Specifically, the contributions of this paper are summarized as follows:

- First, a channel attention module (CAM) and a time attention module (TAM) were designed to enrich the information of related representations. Since different convolution kernels in 1D CNN equal to signal filtering, it could be interpreted as different channels carrying different “frequency” (Hoshen et al., 2015; Sainath et al., 2015), the CAM can explicitly model the features dependency between different convolution channels, and adaptively enhance the discriminant features of related “frequency” and suppress irrelevant ones. Similarly, TAM is designed to enhance the discriminant features of related “time” and avoid the interference from the unrelated frames.

- Second, an joint attention module (TCAM) combining both TAM and CAM is proposed to optimize feature mapping. 1D CNN model can accumulate the benefits of feature recalibration produced by TCAM blocks, and exploits the hidden information of relevance in time and frequency domain simultaneously. By this combining scheme, our model could focus on learning discriminative sound representations instead of those with irrelevant information.
- Last, we introduced a novel snapshot ensemble (Huang et al., 2017) algorithm to further boost our model. Studies in Alamir (2021), Luz et al. (2021), Paseddula and Gangashetty (2021), Sakashita and Aono (2018), Yang et al. (2020), Yin et al. (2018) have shown that ensembling multiple deep networks in audio classification could lead to notable performance improvement, but commonly seen way that training and saving several deep CNNs separately is actually inefficient. With help of cyclic learning rate scheduler, it can extract multiple suboptimal models and ensemble them within a single training process.

We use ESC datasets UrbanSound8K (Salamon et al., 2014) and ESC-10 (Piczak, 2015b) to evaluate the performance of our proposed model. Noted that no data augmentation tricks were used in experiments and our numerical results show: (1) Compared to the top performance of current 1D CNN, our model outperform any of them in terms of accuracy. (2) Our model could produce comparable performance evaluation results with spectrogram-based SOTA 2D CNN model, meanwhile having substantially less parameters. With higher computational efficiency while maintaining high performance, our proposed method have great potentials to be implemented in resource-constrained devices.

The rest of this article is arranged as follows. In Section 2, we explain the basic idea of our proposed one-dimensional convolutional network with designed attention modules, as well as snapshot ensemble algorithm. In Section 3, we describe the datasets, network architecture and experimental configurations. In Section 4, we show experimental results and performance evaluation of our proposed model. Section 5 concludes this study.

2. Methods

2.1. Basics of one-dimensional convolutional neural network

1D CNN model also can be viewed as a sequence of convolutional layers alternating with pooling layers, and followed by dense layers. The key distinction between 2D and 1D CNN is that the 1D model uses one-dimension arrays for both kernels and feature maps.

1D Convolutional layer: Convolution layer is akin to applying filters to sequences, where a kernel window slides over a sequence with specified stride and passes its output to a nonlinear activation function, such as ReLU. Suppose there are number of C kernels in a convolution layer, $\mathbf{F} = [\mathbf{f}^1, \mathbf{f}^2, \mathbf{f}^3, \dots, \mathbf{f}^j, \dots, \mathbf{f}^C]$, then each kernel $\mathbf{f}^j = [f_{j,1}, f_{j,2}, \dots, f_{j,m}]_{(m\times 1)}$ can be viewed as a filter focus on different frequency band. For a t length 1D input $\mathbf{x} = [x_1, x_2, \dots, x_t]_{(t\times 1)}$, after convolution with the jth filter $\mathbf{f}^j$ with a length m, the ith element in the output convoluted feature map is given by

$$c_{j,i} = \sigma\left(\sum_{k=1}^{m} f_{j,m+1-k} \cdot x_{i+k-1} + b_j\right) \tag{1}$$

where σ denotes the activation function, and b_j is a scalar bias. By sliding this window over the entire length of the data, the final feature map corresponding to jth filter can thus be obtained as:

$$\mathbf{c}_j = [c_{j,1}, c_{j,2}, \dots, c_{j,w}]_{(w\times 1)} \tag{2}$$

where w denotes the final length of the obtained convolution feature. For example, assuming $stride = 1$, no-padding, then $w = t - m + 1$.

1D Pooling layer: Pooling layer tries to compress the feature map obtained from previous convolutional layer. Commonly used pooling methods such as max pooling or average pooling is to obtain location-invariant features and reduce the number of parameters as well. Typically, max pooling is frequently employed. With a window size s pooling, compressing the jth filter feature map $\mathbf{c}_j$ to

$$\mathbf{y}_j = [y_{j,1}, y_{j,2}, \dots, y_{j,W}]_{(W\times 1)} \tag{3}$$

where $W = \frac{w}{s}$, $y_{j,k}$ can be expressed as:

$$y_{j,k} = \max\{c_{j,(k-1)s+1}, c_{j,(k-1)s+2}, \dots, c_{j,ks}\} \tag{4}$$

Since the fully connected layer and softmax activation function are identical to 2D CNN, during the training phase, 1D model has the similar conventional back-propagation (BP) formulation in minimizing the cross-entropy loss function between predicted output distribution and the real one.

2.2. Proposed attention models

2.2.1. Channel attention module

Channel Attention Module (CAM) aims to improve the features' discriminative ability by suppressing irrelevant information and adaptively enhancing related activation maps of audio sequence. As Fig. 1 shows, the CAM's input $\mathbf{Y}$ is two dimensional feature generated by passing raw audio sequence through the initial convolution layer. Since initial convolution layer has the number of C kernels and each kernel works like a channel filter, $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_C]$ can be viewed as a collection of frequency information composed of multiple channels. Each channel $\mathbf{y}_i$ is a row vector of size $\mathbb{R}^{1\times W}$.

First, $\mathbf{Y}$ is compressed by global average pooling operation into channel statistical vector $\mathbf{z} \in \mathbb{R}^{1\times C}$, and its ith element z_i is computed by

$$z_i = \text{Avgpool}\left(\mathbf{y}_i\right) = \frac{1}{1\times W}\sum_{j=1}^{W} y_{i,j} \tag{5}$$

Then, the CAM leverages a gating mechanism to capture the inter-channel correlation and generate the recalibrated channel vector $\mathbf{z}'$. The gating mechanism contains two convolutional operations, F' and F'', with a channel number of 1 and a convolution kernel size of 1×1, respectively. To encode dependencies in channel-wise, the output of F' is passed to a ReLU activation function δ, and then the output of F'' is passed to a sigmoid activation function σ, which normalize dynamic value of activation vector into range [0, 1], resulting in a channel recalibration vector $\mathbf{z}'$. Hence, the gating mechanism can be expressed as:

$$\mathbf{z}' = \sigma\left(F''\left(\delta\left(F'(\mathbf{z})\right)\right)\right) \tag{6}$$

Next, the recalibrated feature $\mathbf{M}$ is can be obtained by element-wise multiplication of $\mathbf{Y}$ with $\mathbf{z}'_i$, and its calculation formula is as follows:

$$\begin{aligned}\mathbf{M} &= \left[\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_C\right] = \mathbf{Y} \cdot \mathbf{z}' \\ &= \left[\mathbf{y}_1\mathbf{y}'_1, \mathbf{y}_2 z'_2, \dots, \mathbf{y}_C z'_C\right]\end{aligned} \tag{7}$$

With global information z'_i, $\mathbf{M}$ can effectively highlights the most discriminative channel feature information in $\mathbf{Y}$. A residual connection is also added to improve the optimization feasibility and preserve the original information as well, so the output of CAM is given by

$$\mathbf{Y}_{\text{CAM}} = \mathbf{Y} + \mathbf{M}. \tag{8}$$

2.2.2. Time attention module

The goal of Time Attention Module (TAM) is to locate the important time segments which contain the most related time characteristics of sound category information in audio sequence. As Fig. 2 illustrates, the feature Y can also be viewed as a collection of information composed of multiple time segments and represented by $\left[\mathbf{y}^1, \mathbf{y}^2, \dots, \mathbf{y}^W\right]$, where $\mathbf{y}^j$ is the jth time signal position, $j = 1, 2, \dots, W$.

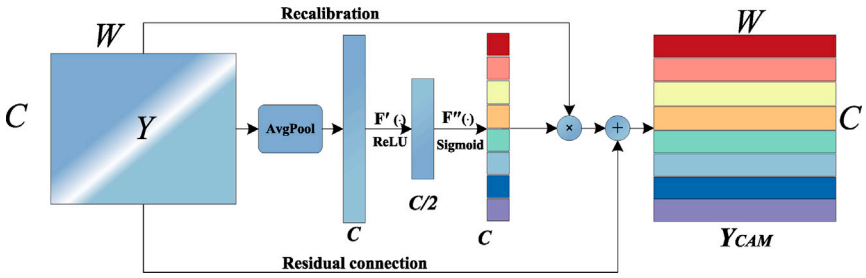


Fig. 1. Basic structure of the CAM.

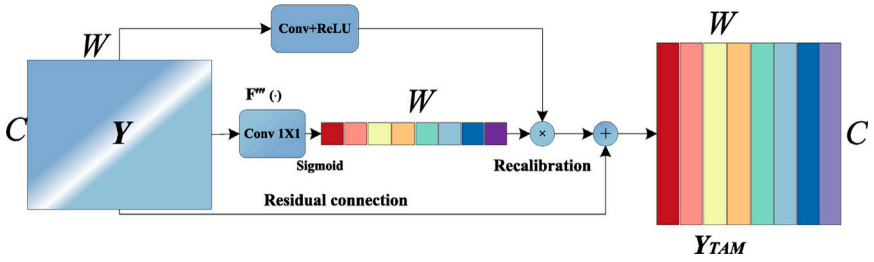


Fig. 2. Basic structure of the TAM.

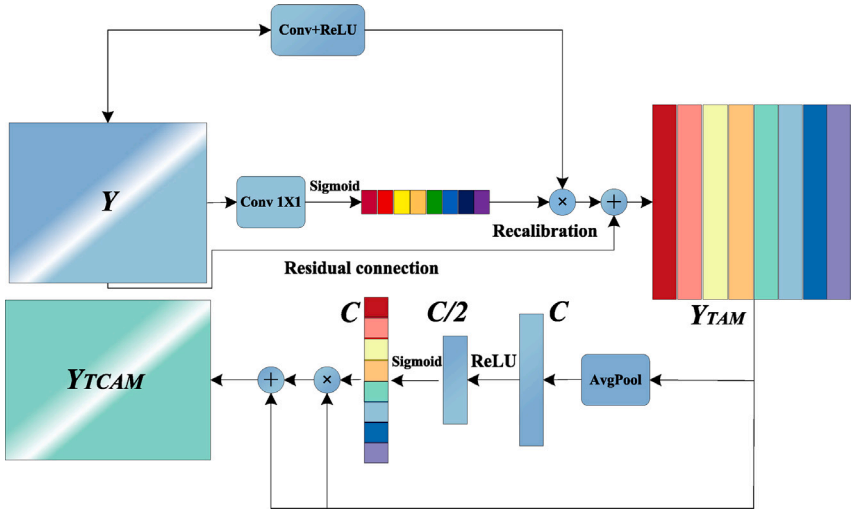


Fig. 3. Basic structure of the Time-Channel Attention Module.

To aggregate all channels features, TAM first projects the feature $\mathbf{Y}$ into a temporal sequence $\mathbf{s}$ by using a 1×1 convolutional layer with a single channel, *i.e.*

$$\mathbf{s} = F'''(\mathbf{Y}) \tag{9}$$

Then use the sigmoid function to get the time weight vector $\mathbf{s}'$:

$$\mathbf{s}' = \sigma(\mathbf{s}), \mathbf{s} \in \mathbb{R}^{1\times W} \tag{10}$$

The $\mathbf{s}'$ serves as an significance indicator of time segments, and can be used to obtain the recalibrated feature map $\mathbf{N}$ by

$$\mathbf{N} = \left[\mathbf{n}^1, \mathbf{n}^2, \ldots, \mathbf{n}^W\right] = F_s(\mathbf{Y}) \cdot \mathbf{s}' \tag{11}$$

where F_s tis a convolutional operation of $\mathbf{Y}$ to prevent $\mathbf{N}$ over-focusing too much on $\mathbf{s}'$.

Similar to CAM, a residual connection is also added to avoid feature response value degradation, so the output of TAM is given by

$$\mathbf{Y}_{\mathrm{TAM}} = \mathbf{Y} + \mathbf{N}. \tag{12}$$

2.2.3. Time-channel attention module

The Time-Channel Attention Module (TCAM) combines TAM and CAM together to adaptively optimize the feature $\mathbf{Y}$ in both channel and time perspectives. As shown in Fig. 3, TCAM recalibrates $\mathbf{Y}$ by channel rescaling and time rescaling, so the final feature map $\mathbf{Y}_{TCAM}$ can be obtained by

$$\mathbf{Y}_{TCAM} = F_{CAM}(F_{TAM}(\mathbf{Y})) \tag{13}$$

If a certain element $Y(i, j)$ in feature map $\mathbf{Y}$ is considered as important by both channel and time recalibration, it will have a much higher chance to be activated in $\mathbf{Y}_{TCAM}$, which results in that the network could more focus on learning discriminative audio features than those are not. It should be noted that this feature recalibration benefits can be accumulated, as such, discrimination of audio features can be further enhanced by TCAM blocks stacking.

2.3. Snapshot ensemble

Snapshot ensemble leverages its local minima escape ability to mitigate the non-convexity issue of neural networks by adjusting the

Table 1
Summary of ESC-10 and UrbanSound8K datasets.

Dataset	classes	# clips	maximum duration of audio clips	Total length	Folds
UrbanSound8K	10	8, 732	≤ 4 s	7.3 h	10
ESC10	10	400	≤ 5 s	0.56 h	5

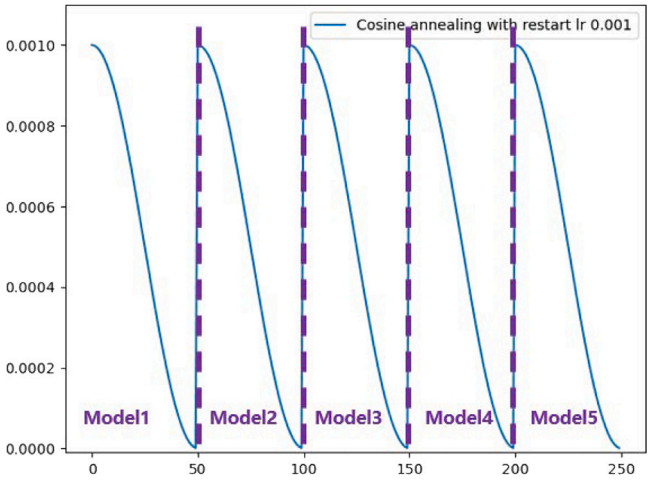


Fig. 4. Cosine annealing learning rate schedule over training epochs.

learning rate (LR) according to a specific schedule during the training phase. In practice, cosine annealing learning rate schedule was implemented to make model ensembling diverse (Loshchilov & Hutter, 2016). This schedule adjusts the learning rate at each epoch t according to the follows equation:

$$\alpha(t) = \frac{\alpha_0}{2}\left(\cos\left(\frac{\pi \mod (t-1, \lceil T/M \rceil)}{\lceil T/M \rceil}\right) + 1\right) \tag{14}$$

where α_0 denotes the maximum learning rate (LR), $\alpha(t)$ represents the LR at epoch t, T and M indicate the number of total cycles, respectively. *mod* is the modulo operation and $\lfloor \cdot \rfloor$ is floor operations.

Typical cosine annealing schedule adjusts the learning rate in a systematically aggressive manner, which results in the generation of diverse model weights across training epochs. As shown in Fig. 4, it periodically decreases learning rate from a high level to a value near zero rapidly and then gradually grows back to the high value again. Since cosine annealing can reach to a local minimum on the performance surface during each cycle, that is to say each cycle's training weights correspond to a sub-optimal model. At test phase, the ensemble prediction can acquire its final classification results by averaging Softmax outputs of top m accurate models out of M in the ensemble training.

$$h_{Ensemble} = \frac{1}{m}\sum_{i=0}^{m-1} h_{M-i}(x) \tag{15}$$

3. Evaluation methodology

Before presenting the experimental results, the basic information of the database, necessary preprocessing, experimental setup and associated evaluation metrics are outlined in this section.

3.1. Database and preparation

Two benchmarked datasets of ESC: UrbanSound8K (Salamon et al., 2014) and ESC-10 (Piczak, 2015b) were used for evaluation in our experiment and information about datasets were summarized in Table 1. For accuracy of evaluation, 10-fold or 5-fold cross-validation scheme were employed, respectively, and we took average of experiments as presented results. Besides, some other necessary processing steps used in experiments are:

Table 2
Architectures of proposed multiattention one-dimensional convolutional neural network for environmental sound classification.

Layer	Type	Kernel	Channel	Stride	Padding	Output
0	Input	–	–	–	–	8000 × 1
1	Conv	32 × 1	32	1	Yes	8000 × 32
2	TAM	1 × 1	1	–	Yes	8000 × 32
3	CAM	–	–	–	–	8000 × 32
4	Conv	16 × 1	32	2	Yes	4000 × 32
5	TAM	1 × 1	1	–	Yes	4000 × 32
6	CAM	–	–	–	–	4000 × 32
7	Conv	9 × 1	64	2	Yes	2000 × 64
8	TAM	1 × 1	1	–	Yes	2000 × 64
9	CAM	–	–	–	–	2000 × 64
10	Conv	6 × 1	64	2	Yes	1000 × 64
11	TAM	1 × 1	1	–	Yes	1000 × 64
12	CAM	–	–	–	–	1000 × 64
13	Conv	3 × 1	128	5	Yes	200 × 128
14	TAM	1 × 1	1	–	Yes	200 × 128
15	CAM	–	–	–	–	200 × 128
16	Conv	3 × 1	128	5	Yes	40 × 128
17	TAM	1 × 1	1	–	Yes	40 × 128
18	CAM	–	–	–	–	40 × 128
19	Conv	3 × 1	256	2	Yes	20 × 256
Global average pooling						
Softmax						
	C1 C2	C3 C4	C5 C6	C7 C8	C9 C10	

- **Data resampling:** To unify the shape of the input, each audio sample was resampled to 16 kHz and zeropadded in the end to reach 4 s or 5 s length if necessary.
- **Data framing:** An appropriate width sliding widow (Abdoli et al., 2019) was employed to split the audio samples into frames. These frames overlap in a certain percentage to optimize the utilization of audio information.
- **Frames aggregation**: Multiple predictions combination is needed in order to carry out an effective classification, here the majority vote and sum rule (Abdoli et al., 2019) were adopted and compared in the follow-up experiments.

3.2. Proposed network architecture

Table 2 shows the overall architecture of our 1D CNN model with proposed TCAM module. Inspired by multi-head attention, it consists of a backbone network, one convolutional layer and a classification layer. The backbone cascades six convolutional modules in series, each convolutional module contains a convolutional layer with a ReLU function and concatenated by a TCAM which can adaptively perform feature recalibrations. The classification layer substitutes conventional full connection with a global average pooling to address the overfitting issue, and followed by a softmax function such that the final classification can be carried out. Note that, the appropriate length of the input of the first layer is obtained by subsequent experimental evaluation and the output layer is determined according to the classification category. Here, we take the input length 8000 and 10 categories as an example.

3.3. Evaluation setup

- **Experimental environment:** the proposed networks are developed in Python and implemented with Tensorflow framework.

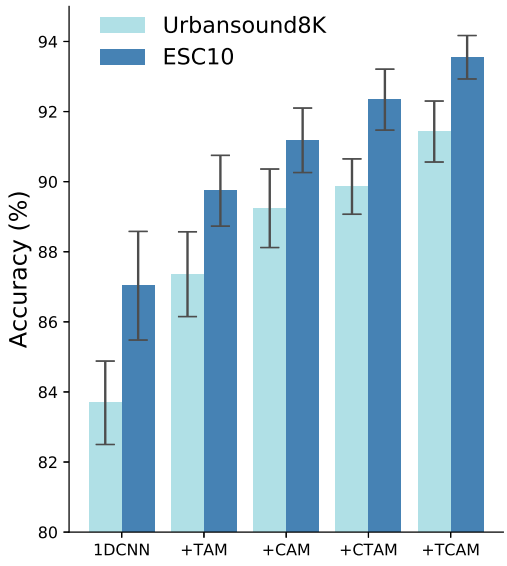


Fig. 5. Results of different attention combinations.

Table 3
Performance of the proposed models under different parameter configuration: frame length (8000, 16000), overlap ratio (0%, 25%, 50%, and 75%) and decision fusion (SUM, MajorVoting(MAJ)).

Frame lenghth	Overlap ratio	UrbanSound8K		ESC10	
		SUM	MAJ	SUM	MAJ
16000	0.	89.2%	87.8%	87.5%	85.0%
	0.25	89.9%	88.2%	87.5%	86.25%
	0.50	90.3%	87.1%	90.0%	87.5%
	0.75	91.8%	89.5%	90.0%	85.0%
8000	0.	88.7%	86.8%	87.5%	85.0%
	0.25	89.2%	87.2%	87.5%	85.0%
	0.50	91.4%	89.2%	90.0%	87.5%
	0.75	90.8%	87.4%	87.5%	85.0%

The code was run in GeForce RTX 3090 Graphics Card with 24 GB memory under Ubuntu 20.04.1 LTS. Training was performed using Adam optimization with learning rate set to 0.0002. The batch size was 100 and mean squared logarithmic error was chosen as the loss function. The popular Glorot initialization was employed to initialize the network weights.[1]

- **Evaluation metrcis:** We adopt the well-known accuracy metric for performance evaluation, which was defined as:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} * 100\% \tag{16}$$

where TP, FP, TN and FN represent true positive samples, false positive samples, true negative samples, and false negative samples respectively. Additionally, the number of network parameters were also compared to show the computational efficiency of proposed 1D CNN model. As experiment results will demonstrate, evaluation metrics show that our proposed model is well-fitted for ESC in a comprehensive way.

4. Experimental results

4.1. Evaluation on frame length, overlap and aggregation strategies

The goal of the initial experiments is to find the optimal configuration for the proposed 1D CNN network. The model in Table 2 is evaluated on two different frame lengths (8000, 16000) with four overlapping percentages (0%, 25%, 50%, and 75%), and two frame aggregation strategies (majority voting or the sum rule) were adopted in experiments. The experimental results are summarized in Table 3.

It can be seen from Table 3 that sum rule significantly improved performance in all cases compared to majority voting, and 16000-frame length with 75% overlap achieve the best accuracy across two datasets. It should be noted that even though the 8000-frame length with 50% overlap does not achieve best performance, the half reduced parameters with almost same accuracy make it more practical than the other configurations. Hence, considering the model performance and resource-constrained devices in realistic, instead, the 8000-frame length with 50% overlap under SUM voting strategy is chosen for the subsequent experiments.

4.2. Evaluation on attention module configuration

Attention combination scheme: To have a further insight of performance influence caused by proposed attention modules, we evaluated various combination schemes. Specifically, model substitutes corresponding TCAM layer in Table 3 with some other modules listed as follows:

- 1D CNN without attention module;
- 1D CNN with TAM only;
- 1D CNN with CAM only;
- 1D CNN with CTAM (swapping inner combining order of the TAM/CAM module);

The experimental results were listed in Fig. 5.

Obviously, no matter which modules 1D CNN chooses to build the network, the accuracy of vanilla 1D CNN can be easily outperformed. It proves that both of the proposed two attention modules can effectively improve the discriminative ability of extracting feature. Also, we found that the performance improvement brought by CAM is higher than TAM, and that of TCAM is higher than CTAM. This is because CAM pays more attention to learn frequency-related discriminative features and learning sound information from "frequency" is more efficient than "time". As for the latter, since TAM in CTAM module first uses convolution to compress all channels before recalibration, it loses frequency information in some extend and eventually decreases its model performance. Overall, two attention modules CAM and TAM are complementary, recalibration in both frequency and time aspects can make features more discriminative. Since attention combination scheme does affect the accuracy, the TCAM Scheme is adopted for 1D CNN in the following experiments.

Number of TCAM stacked: The optimal stacking number of TCAM for the proposed 1D CNN model was investigated, the results are summarized in Fig. 6. It can be observed by both datasets curves that the accuracy of classification keep rising with number of TCAM stacked from 1 to 6, but begin to drop when it reach 7. This proves that feature recalibration benefits of proposed TCAM can be accumulated (before 6) and 1D CNN performance can be improved progressively by TCAM stacking, but this positive effect starts to diminish at certain threshold (7 in experiment) indicates that too many TCAM stacked may cause feature over-recalibrated, which results in accuracy drop. Thus, the number of TCAM modules stacked in Table 3 is selected as 6.

4.3. Evaluation on snap ensemble

Learning rate and number of cycles are two main parameters that affect snap ensemble behavior, we investigated these parameters separately to find the optimal configuration of snap ensemble for proposed model. Noted that the total epochs in experiment train budget T is experimentally set to 200 epoch. Since snapshot models from later cycles received more training and more likely to converge into a local

[1] In our experiments with our model, initialized with Gammatone filterbanks (Abdoli et al., 2019), we did not observe any improvement.

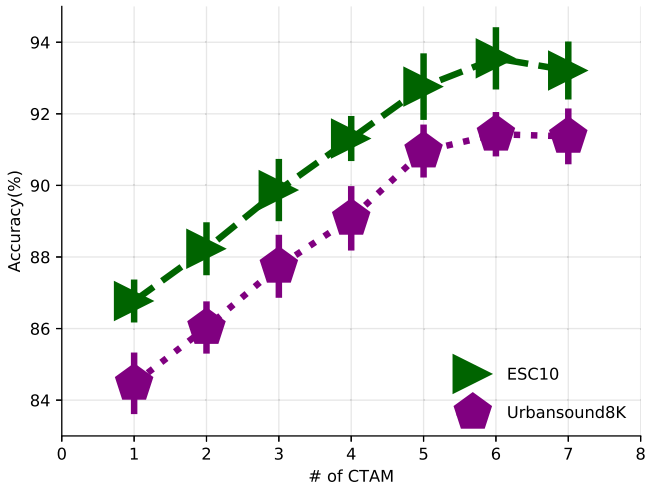


Fig. 6. Results of different number of attention modules.

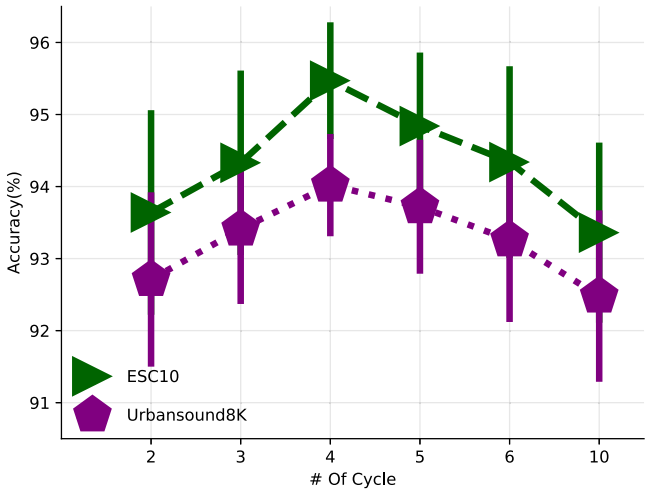


Fig. 8. Results with different number of cycles.

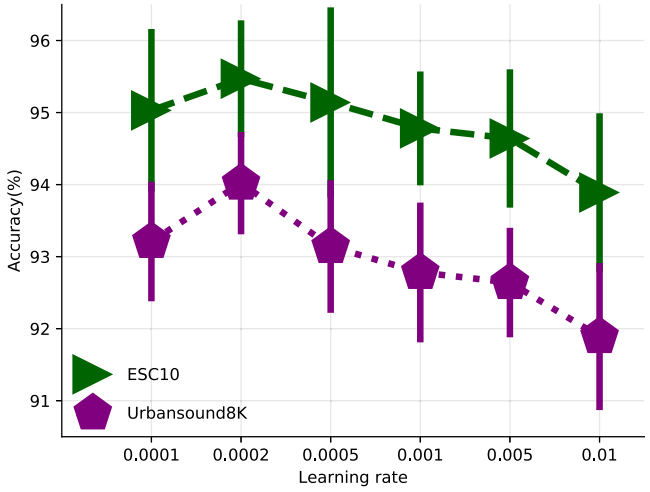


Fig. 7. Results with different learning rate.

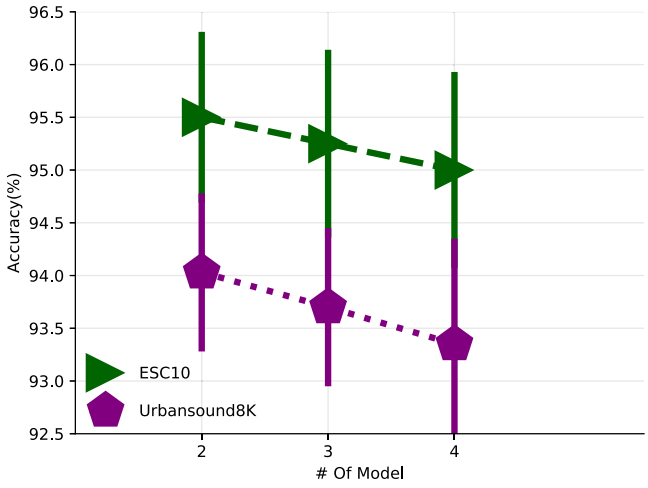


Fig. 9. Results with different ensemble size.

minimum with a better performance (Huang et al., 2017), we choose the last two snapshot models' outputs as final prediction.

Learning rate scheme: We first choose a moderate value 4 for annealing cycles, that is to say each cycle has 200/4 = 50 epochs. Fig. 7 summarized experimental results of six different learning rates, it is obvious that snap ensemble can improve the accuracy regardless of learning rate level, but different learning rates affect the performance to different degrees. The reason lies on the fact that a large learning rate causes model converging too fast, high chance jumping over from global minimum into a local minimum. Conversely, small learning rate leads to weak perturbation, low chance jumping into another local minimum, which reduces the diversity of suboptimal solutions. It can be seen that a appropriate learning rate 0.0002 can help model improve the accuracy most.

Number of cycles: Given a fixed training budget T, the best number of cycles can be find by numerical search. Here the number of cycles is gradually increased from 2 to 10 and the corresponding results are presented in Fig. 8. It can be observed that, no matter what value the number of cycles was set to, it outperforms non-ensemble model by a large margin, but with different degrees. The large number means a few epochs, which results in insufficient training. On the other hand, small number produces less available snapshot models, but later cycles' models are more likely to be high performance, which means some trade-off need be made. Hence, the number of cycles should not be set too large or too small. As the figure shows, setting number of cycles to 4 obtains the best accuracy.

Ensemble size: In practice, ensemble size depends on the available computation resource of deployed devices, more models to ensemble, more computation resource will be needed. Fig. 9 shows our proposed model accuracy versus ensemble size with optimal configuration set from previous experiments. The curves obviously shows that ensemble of two snapshot models already achieves optimal condition, adding more snapshot models result in a performance drop. The reason is because accuracy and diversity are equally important for ensembling, in this regard, for training on a limited epochs budget, these may not be guaranteed.

4.4. Complexity versus accuracy

In addition to accuracy, complexity of model is another important factor that needs to be considered in practical application, which is mainly reflected by the number of parameters of network. Table 4 summarized our proposed model compare results with other state-of-the-art works in literature, since data augmentation was not adopted in our method, those models that trained with data augmentation are marked out.

We can observe from top half of the table that the proposed model achieves the highest accuracy (94.04% for UrbanSound8K and 95.5%

Table 4
The accuracy and computational complexity of the proposed model and other existing ESC models. "Params" denotes the number of model parameters and "FLOPs" denotes the number of floating point operations. UB8K: UrbanSound8K, -: Not available, DA: With data augmentation, SE: snapshot ensemble, MFCC: Mel frequency cepstral coefficient, MelSpec: Mel spectrogram, GamSpec: gammatone spectrogram, LogMelSpec: log-scaled mel-spectrogram , LogGamSpec: log-scaled gammatone spectrogram, CST: chroma, spectral contrast and tonnetz.

Methods	Model input	UB8K	ESC10	Params	FLOPs
TCAM1DCNN (Proposed single model)	Raw waveform	91.43%	93.50%	406 K	40 M
SE-TCAM1DCNN (Proposed two models ensembled)	Raw waveform	94.04%	95.50%	~810 K	~80 M
Gupta's Model (DA) (Gupta et al., 2022)	Raw waveform	86%	80.40%	1.6 M	160 M
Bayesian Optimization Ensemble 1DCNN (five models) (Ragab et al., 2021)	Raw waveform	94%	–	1.9 M	214.7 M
TimeScaleNet (Bavu et al., 2019)	Raw waveform	–	69.71%	10.7 M	–
1DCNN Gamma (Abdoli et al., 2019)	Raw waveform	89%	–	550 K	–
RawNet (Li et al., 2018)	Raw waveform	87.7%	85.2%	377 K	–
EnvNet-v2 (Tokozume & Harada, 2017)	Raw waveform	78%	90%	101 M	–
M18 CNN (Dai et al., 2017)	Raw waveform	72%	–	3.7 M	–
CNN optimization with PSO (İnik, 2023)	Scalogram	91.17%	88.50%	–	–
CNN optimization with PSO (DA) (İnik, 2023)	Scalogram	98.45%	98.64%	–	–
Self-attention transformer (Song et al., 2022)	Scattering Transform	97.7%	–	–	–
Gupta's Model (DA) (Gupta et al., 2022)	GamSpec	89%	80.20%	1.8 M	6.96 G
CNN-RNN (DA) (Bahmei et al., 2022)	MelSpec	98%	–	20.5 M	–
Attention conlutional RNN (Zhang et al., 2021)	LogGamSpec	–	91.7%	3.81 M	9.18 M
Attention conlutional RNN (DA) (Zhang et al., 2021)	LogGamSpec	–	93.7%	3.81 M	9.18 M
ESResNet (Palanisamy et al., 2020)	LogMelSpec	85.4%	97.0%	23.61 M	183.36 G
Regularization DCNN (Mushtaq & Su, 2020)	LogMelSpec	94.14%	81.25%	3.17 M	61.03 M
Regularization DCNN (DA) (Mushtaq & Su, 2020)	LogMelSpec	95.37%	94.94%	3.17 M	61.03 M
MCLNN (Medhat et al., 2020)	LogMelSpec	74.22%	85.5%	1.2 M	–
6-layer CNN (Su et al., 2020)	MFCC+LogMelSpec+CST	93.4%	–	11.3 M	–
Dilated CNN (Chen et al., 2019)	LogMelSpec	78%	–	1.56 M	–
TSCNN-DS (Su et al., 2019)	MFCC+LogMelSpec+CST	97.2%	–	15.9 M	–
VGG (Pons & Serra, 2019)	LogMelSpec	70%	–	77 M	–
SB-CNN(DA) (Salamon & Bello, 2017)	LogMelSpec	79%	–	241 K	–
AlexNet/GoogLeNet (Boddapati et al., 2017)	Spectrogram	92% /93%	–	60 M/6.8 M	–
PiczakCNN (DA) (Piczak, 2015a)	LogMelSpec	73%	80%	31.53 M	63.27 M

for ESC10) in 1D CNN category, and the number of its parameters (810 K for ensembling) is still less than 1M, which is the lowest level as RawNet (Li et al., 2018) but with much higher accuracy. As for the comparison results with 2D CNN category, it can be seen from bottom half of the table that, among all the top accuracy (above 90%) models, our TCAM1DCNN has significant improvement in model complexity. It is at least 2 times less than CNN-RNN (Bahmei et al., 2022) and almost 20 times less than TSCNN-DS (Su et al., 2019). Furthermore, the FLOPs of our TCAM1DCNN is merely 40 M, which achieves the lowest level of float operations requirement among all the available 1D CNN models. Even consider SE-TCAM1DCNN whose FLOPs are doubled, our proposed model still holds lowest value. As for the comparison with 2D CNN models, although both Attention convolutional RNNs merely have FLOPs of 9.18 M, but their model size and accuracy are not as good as ours.

Overall, the numerical results demonstrated that our proposed method could achieve SOTA level performance in terms of accuracy while significantly reduce model complexity. As mentioned before, there is no data augmentation trick used in our model, which makes it well suited for applications with limited computational resources, such as remote IoT devices deployment for ESC.

It needs to be pointed out that Table 4 does not involve literature about transformer models. The reason why they are excluded is that, the goal of our work is to propose a lightweight 1D CNN model which can be easily implemented in resource limited deceives, meanwhile achieve competitive accuracy with 2D SOTA CNN models. Most of these comparison CNN models are tested on UB8K and ESC10, but for current transformer models (Atito et al., 2022; Chen, Du, et al., 2022; Chen, Wu, et al., 2022; Gong et al., 2021; Koutini et al., 2021; Li & Li, 2022; Liu & Fang, 2023; Liu et al., 2023; Ristea et al., 2022; Zhang et al., 2022; Zhu & Omar, 2023), since they are usually designed to do more complicated ESC tasks, the test datasets often contain many target sound types, such as ESC50 and Audioset, which are more complicated than UB8K and ESC10. On the other hand, those transformer often possess relatively larger model size (parameters are above 30 M in average and with FLOPs more than 20 G) and their accuracy are about 96%. Without same test benchmark, simply put those accuracy or model size together in Table 4 is not a fair comparison.

5. Conclusion and future works

With lower computational complexity, no extra pre-feature extraction, 1D CNN based on raw audio waveform for environment sound classification are receiving growing attention. Nevertheless, compared to the SOTA 2D CNN models, currently there still exists a large performance gap. In this study, we introduce two attention modules and combined them both to recalibrate the intermediate feature maps in both time and channel aspects, leveraging it to adaptively focus on informative features and suppress less useful ones with the help of global information. Furthermore, a novel snapshot ensemble, without incurring any additional training costs, is also incorporated to further boost performance.

With proposed attention modules, extensive comparison were conducted on two public benchmark ESC datasets. In compassion to other 1D CNN models, our model brings considerable performance improvement, which highlights the effectiveness and importance of proposed attention mechanism. In compassion to other 2D CNN models, the performance is close to the reported best accuracy while notably requiring less significant parameters. Although its accuracy did not exceed the highest 2D CNN model, the good balance of efficiency and accuracy indicates its promising in ESC application with power and hardware-constrained IoT devices.

CRediT authorship contribution statement

Huaxing Xu: Conceptualization, Methodology, Software, Formal analysis, Writing – original draft, Writing – review & editing, Funding acquisition. **Yunzhi Tian:** Data curation, Software, Formal analysis, Writing – original draft. **Haichuan Ren:** Resources, Validation, Writing – review. **Xudong Liu:** Validation, Supervision, Writing – review & editing.

Declaration of competing interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our works.

Data availability

The authors do not have permission to share data.

Acknowledgments

This work was partially supported by National Key R&D Program of China (Grant No. 2022YFC3502400) and National Natural Science Foundation of China (Grant No. 11804309).

References

Abdoli, S., Cardinal, P., & Koerich, A. L. (2019). End-to-end environmental sound classification using a 1D convolutional neural network. *Expert Systems with Applications*, *136*, 252–263.

Alamir, M. A. (2021). A novel acoustic scene classification model using the late fusion of convolutional neural networks and different ensemble classifiers. *Applied Acoustics*, *175*, Article 107829.

Atito, S., Awais, M., Wang, W., Plumbley, M. D., & Kittler, J. (2022). ASiT: Audio spectrogram vision transformer for general audio representation. arXiv preprint: 2211.13189.

Aucouturier, J. J., Defreville, B., & Pachet, F. (2007). The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music. *The Journal of the Acoustical Society of America*, *122*(2), 881–891.

Aytar, Y., Vondrick, C., & Torralba, A. (2016). Soundnet: Learning sound representations from unlabeled video. *Advances in Neural Information Processing Systems*, *29*.

Bahmei, B., Birmingham, E., & Arzanpour, S. (2022). CNN-RNN and data augmentation using deep convolutional generative adversarial network for environmental sound classification. *IEEE Signal Processing Letters*, *29*, 682–686.

Bavu, E., Ramamonjy, A., Pujol, H., & Garcia, A. (2019). TimeScaleNet: A multiresolution approach for raw audio recognition using learnable biquadratic IIR filters and residual networks of depthwise-separable one-dimensional atrous convolutions. *IEEE Journal of Selected Topics in Signal Processing*, *13*(2), 220–235.

Bisot, V., Serizel, R., Essid, S., & Richard, G. (2017). Feature learning with matrix factorization applied to acoustic scene classification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *25*(6), 1216–1229.

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

Chandrakala, S., & Jayalakshmi, S. (2019). Environmental audio scene and sound event recognition for autonomous surveillance: A survey and comparative studies. *ACM Computing Surveys*, *52*(3), 1–34.

Chen, K., Du, X., Zhu, B., Ma, Z., Berg-Kirkpatrick, T., & Dubnov, S. (2022). HTS-AT: A hierarchical token-semantic audio transformer for sound classification and detection. In *ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing* (pp. 646–650). IEEE.

Chen, Y., Guo, Q., Liang, X., Wang, J., & Qian, Y. (2019). Environmental sound classification with dilated convolutions. *Applied Acoustics*, *148*, 123–132.

Chen, S., Wu, Y., Wang, C., Liu, S., Tompkins, D., Chen, Z., & Wei, F. (2022). Beats: Audio pre-training with acoustic tokenizers. arXiv preprint: 2212.09058.

Chu, S., Narayanan, S., & Kuo, C.-C. J. (2009). Environmental sound recognition with time–frequency audio features. *IEEE Transactions on Audio, Speech, and Language Processing*, *17*(6), 1142–1158.

Crocco, M., Cristani, M., Trucco, A., & Murino, V. (2016). Audio surveillance: A systematic review. *ACM Computing Surveys*, *48*(4), 1–46.

Dai, W., Dai, C., Qu, S., Li, J., & Das, S. (2017). Very deep convolutional neural networks for raw waveforms. In *2017 IEEE international conference on acoustics, speech and signal processing* (pp. 421–425). IEEE.

Dhanalakshmi, P., Palanivel, S., & Ramalingam, V. (2009). Classification of audio signals using SVM and RBFNN. *Expert Systems with Applications*, *36*(3), 6069–6075.

Eronen, A. J., Peltonen, V. T., Tuomi, J. T., Klapuri, A. P., Fagerlund, S., Sorsa, T., Lorho, G., & Huopaniemi, J. (2005). Audio-based context recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, *14*(1), 321–329.

Gong, Y., Chung, Y.-A., & Glass, J. (2021). Ast: Audio spectrogram transformer. arXiv preprint: 2104.01778.

Gupta, S. S., Hossain, S., & Kim, K.-D. (2022). Recognize the surrounding: Development and evaluation of convolutional deep networks using gammatone spectrograms and raw audio signals. *Expert Systems with Applications*, *200*, Article 116998.

Hoshen, Y., Weiss, R. J., & Wilson, K. W. (2015). Speech acoustic modeling from raw multichannel waveforms. In *2015 IEEE international conference on acoustics, speech and signal processing* (pp. 4624–4628). IEEE.

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).

Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J. E., & Weinberger, K. Q. (2017). Snapshot ensembles: Train 1, get m for free. arXiv preprint arXiv:1704.00109.

Huzaifah, M. (2017). Comparison of time-frequency representations for environmental sound classification using convolutional neural networks. arXiv preprint arXiv: 1706.07156.

İnik, Ö. (2023). CNN hyper-parameter optimization for environmental sound classification. *Applied Acoustics*, *202*, Article 109168.

Kong, Q., Cao, Y., Iqbal, T., Wang, Y., Wang, W., & Plumbley, M. D. (2020). Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *28*, 2880–2894.

Koutini, K., Schlüter, J., Eghbal-Zadeh, H., & Widmer, G. (2021). Efficient training of audio transformers with patchout. arXiv preprint: 2110.05069.

Kumar, A., & Ithapu, V. (2020). A sequential self teaching approach for improving generalization in sound event recognition. In *International conference on machine learning* (pp. 5447–5457). PMLR.

Kumar, A., Khadkevich, M., & Fügen, C. (2018). Knowledge transfer from weakly labeled audio using convolutional neural network for sound events and scenes. In *2018 IEEE international conference on acoustics, speech and signal processing* (pp. 326–330). IEEE.

Li, X., & Li, X. (2022). ATST: Audio representation learning with teacher-student transformer. arXiv preprint: 2204.12076.

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*(7), 1152.

Liu, F., & Fang, J. (2023). Multi-scale audio spectrogram transformer for classroom teaching interaction recognition. *Future Internet*, *15*(2), 65.

Liu, X., Lu, H., Yuan, J., & Li, X. (2023). CAT: Causal audio transformer for audio classification. In *ICASSP 2023-2023 IEEE international conference on acoustics, speech and signal processing* (pp. 1–5). IEEE.

Loshchilov, I., & Hutter, F. (2016). Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983.

Luz, J. S., Oliveira, M. C., Araujo, F. H., & Magalhães, D. M. (2021). Ensemble of handcrafted and deep features for urban sound classification. *Applied Acoustics*, *175*, Article 107819.

Medhat, F., Chesmore, D., & Robinson, J. (2020). Masked conditional neural networks for sound classification. *Applied Soft Computing*, *90*, Article 106073.

Mushtaq, Z., & Su, S. F. (2020). Environmental sound classification using a regularized deep convolutional neural network with data augmentation. *Applied Acoustics*, *167*, Article 107389.

Nguyen, T., & Pernkopf, F. (2018). Acoustic scene classification using a convolutional neural network ensemble and nearest neighbor filters. In *DCASE* (pp. 34–38).

Ntalampiras, S., Potamitis, I., & Fakotakis, N. (2009). On acoustic surveillance of hazardous situations. In *2009 IEEE international conference on acoustics, speech and signal processing* (pp. 165–168). IEEE.

Palanisamy, K., Singhania, D., & Yao, A. (2020). Rethinking CNN models for audio classification. arXiv preprint arXiv:2007.11154.

Paseddula, C., & Gangashetty, S. V. (2021). Late fusion framework for acoustic scene classification using LPCC, SCMC, and log-Mel band energies with deep neural networks. *Applied Acoustics*, *172*, Article 107568.

Pham, L., Phan, H., Nguyen, T., Palaniappan, R., Mertins, A., & McLoughlin, I. (2021). Robust acoustic scene classification using a multi-spectrogram encoder-decoder framework. *Digital Signal Processing*, *110*, Article 102943.

Phan, H., Hertel, L., Maass, M., Koch, P., Mazur, R., & Mertins, A. (2017). Improved audio scene classification based on label-tree embeddings and convolutional neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *25*(6), 1278–1290.

Phan, H., Hertel, L., Maass, M., Mazur, R., & Mertins, A. (2016). Learning representations for nonspeech audio events through their similarities to speech patterns. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *24*(4), 807–822.

Piczak, K. J. (2015a). Environmental sound classification with convolutional neural networks. In *2015 IEEE 25th international workshop on machine learning for signal processing* (pp. 1–6). IEEE.

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

Pons, J., & Serra, X. (2019). Randomly weighted cnns for (music) audio classification. In *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing* (pp. 336–340). IEEE.

Purwins, H., Li, B., Virtanen, T., Schlüter, J., Chang, S.-Y., & Sainath, T. (2019). Deep learning for audio signal processing. *IEEE Journal of Selected Topics in Signal Processing*, *13*(2), 206–219.

Ragab, M. G., Abdulkadir, S. J., Aziz, N., Alhussian, H., Bala, A., & Alqushaibi, A. (2021). An ensemble one dimensional convolutional neural network with Bayesian optimization for environmental sound classification. *Applied Sciences*, *11*(10), 4660.

Ristea, N., Ionescu, R., & Khan, F. (2022). SepTr: Separable transformer for audio spectrogram processing. arxiv 2022. arXiv preprint: 2203.09581.

Sainath, T., Weiss, R. J., Wilson, K., Senior, A. W., & Vinyals, O. (2015). Learning the speech front-end with raw waveform CLDNNs.

Sakashita, Y., & Aono, M. (2018). Acoustic scene classification by ensemble of spectrograms based on adaptive temporal divisions. *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*.

Salamon, J., & Bello, J. P. (2017). Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, *24*(3), 279–283.

Salamon, J., Jacoby, C., & Bello, J. P. (2014). A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on multimedia* (pp. 1041–1044).

Sigtia, S., Stark, A. M., Krstulović, S., & Plumbley, M. D. (2016). Automatic environmental sound recognition: Performance versus computational cost. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *24*(11), 2096–2107.

Singh, V. K., Sharma, K., & Sur, S. N. (2023). A survey on preprocessing and classification techniques for acoustic scene. *Expert Systems with Applications*, Article 120520.

Song, S., Zhang, C., & Wei, Z. (2022). Research on scattering transform of urban sound events detection based on self-attention mechanism. *IEEE Access*, *10*, 120804–120822.

Su, Y., Zhang, K., Wang, J., & Madani, K. (2019). Environment sound classification using a two-stream CNN based on decision-level fusion. *Sensors*, *19*(7), 1733.

Su, Y., Zhang, K., Wang, J., Zhou, D., & Madani, K. (2020). Performance analysis of multiple aggregated acoustic features for environment sound classification. *Applied Acoustics*, *158*, Article 107050.

Tokozume, Y., & Harada, T. (2017). Learning environmental sounds with end-to-end convolutional neural network. In *2017 IEEE international conference on acoustics, speech and signal processing* (pp. 2721–2725). IEEE.

Tokozume, Y., Ushiku, Y., & Harada, T. (2018). Learning from between-class examples for deep sound recognition. arXiv preprint arXiv:1711.10282.

Tripathi, A. M., & Paul, K. (2022). Data augmentation guided knowledge distillation for environmental sound classification. *Neurocomputing*, *489*, 59–77.

Valero, X., & Alias, F. (2012). Gammatone cepstral coefficients: Biologically inspired features for non-speech audio classification. *IEEE Transactions on Multimedia*, *14*(6), 1684–1689.

Virtanen, T., Plumbley, M. D., & Ellis, D. (2018). *Computational analysis of sound scenes and events*. Springer.

Vuegen, L., Broeck, B., Karsmakers, P., Gemmeke, J. F., Vanrumste, B., & Hamme, H. (2013). An MFCC-GMM approach for event detection and classification. In *IEEE workshop on applications of signal processing to audio and acoustics* (pp. 1–3).

Wang, H., Liu, Z., Peng, D., & Qin, Y. (2019). Understanding and learning discriminant features based on multiattention 1DCNN for wheelset bearing fault diagnosis. *IEEE Transactions on Industrial Informatics*, *16*(9), 5735–5745.

Wu, B., & Zhang, X. P. (2021). Environmental sound classification via time–frequency attention and framewise self-attention-based deep neural networks. *IEEE Internet of Things Journal*, *9*(5), 3416–3428.

Yang, L., Tao, L., Chen, X., & Gu, X. (2020). Multi-scale semantic feature fusion and data augmentation for acoustic scene classification. *Applied Acoustics*, *163*, Article 107238.

Yin, Y., Shah, R. R., & Zimmermann, R. (2018). Learning and fusing multimodal deep features for acoustic scene categorization. In *Proceedings of the 26th ACM international conference on multimedia* (pp. 1892–1900).

Zeinali, H., Burget, L., & Cernocky, J. (2018). Convolutional neural networks and x-vector embedding for DCASE2018 acoustic scene classification challenge. arXiv preprint arXiv:1810.04273.

Zhang, P., Chen, H., Bai, H., & Yuan, Q. (2019). Deep scattering spectra with deep neural networks for acoustic scene classification tasks. *Chinese Journal of Electronics*, *28*(6), 1177–1183.

Zhang, Y., Li, B., Fang, H., & Meng, Q. (2022). Spectrogram transformers for audio classification. In *2022 IEEE international conference on imaging systems and techniques* (pp. 1–6). IEEE.

Zhang, Z., Xu, S., Cao, S., & Zhang, S. (2018). Deep convolutional neural network with mixup for environmental sound classification. In *Chinese conference on pattern recognition and computer vision* (pp. 356–367). Springer.

Zhang, Z., Xu, S., Zhang, S., Qiao, T., & Cao, S. (2021). Attention based convolutional recurrent neural network for environmental sound classification. *Neurocomputing*, *453*, 896–903.

Zhu, W., & Omar, M. (2023). Multiscale audio spectrogram transformer for efficient audio classification. In *ICASSP 2023-2023 IEEE international conference on acoustics, speech and signal processing* (pp. 1–5). IEEE.