Environmental Sound Classification Method Based on Two-Stream Lightweight Convolutional Neural Network

Jingru Fang
Ocean University of China
Bo Yin (ybfirst@ouc.edu.cn)
Ocean University of China
Xiaopeng Ji
Ocean University of China
Zehua Du
Ocean University of China

Research Article

Keywords: environmental sound, TSLCNN-DS, mechanism and residual learning, UrbanSound8k

Posted Date: September 3rd, 2021

DOI: https://doi.org/10.21203/rs.3.rs-860631/v1

Environmental Sound Classification Method Based on Two-Stream Lightweight Convolutional Neural Network

Jingru Fang [1] • Bo Yin [1, 2, *] • Xiaopeng Ji [1] • Zehua Du [1]

[1] College of Information Science and Engineering, Ocean University of China, Qingdao, 266100, China

[2] Pilot National Laboratory for Marine Science and Technology, Qingdao 266237, China

*Correspondence: ybfirst@ouc.edu.cn

Abstract

Neural networks have achieved success in the task of environmental sound classification. However, the traditional neural network model has too many parameters and high computational cost. The lightweight networks solve these problems by compressing parameters, but reduce the classification accuracy. To solve the problems in existing research, we propose a two-stream model based on two lightweight convolutional neural networks, called TSLCNN-DS, which saves memory and improves the classification performance of environmental sounds. Specifically, we first used data patching and data balancing to slightly expand the amount of experimental data. Then we designed two lightweight and efficient classification networks based on the attention mechanism and residual learning. Finally, the Dempster-Shafer evidence theory is used to fuse the output of the two networks, and the two-stream model is integrated. Experiments have shown that the model has achieved a classification accuracy of 97.44% on the UrbanSound8k dataset, using only 0.12 M parameters.

Introduction

In recent years, audio recognition technology has developed rapidly and has been applied in various fields. The more popular ones are voice recognition of different animals [1] speaker recognition [2], and music genre classification [3]. More advanced applications include voice-controlled robots [4], recognizing the heart rate of patients with pulmonary hypertension and so on [5]. However, signals such as voice and music have a specific audio scene or structure. The environmental sound is non-structural, with strong complexity and noise interference. Therefore, environmental sound classification still faces many difficulties.

To cope with these problems, many machine learning models and various feature extraction techniques have been applied to environmental sound classification. In the initial research, people used traditional machine learning methods such as support vector machine (SVM) [6] and Gaussian mixture model [7] to solve the ESC problem. However, these traditional machine learning methods are not suitable for training large-scale samples. Recently, with the development of deep learning methods in the field of pattern recognition, CNNs have also been widely used in voice recognition systems [8, 9, 10, 11, 13]. We divide the research of CNN in environmental sound recognition into three categories.

In the first type of method, the study uses a single audio data as input. Piczak [14] first proposed the application of two-dimensional CNN to learn log-mel spectrogram features, and the accuracy rate on UrbanSound8k was 72.7%. Dai [12] et al. proposed a one-dimensional CNN with 34 layers, directly using time-domain waveforms as input. It is proved that the deep convolutional neural network can obtain better results. This method saves the steps of manually extracting features. However, 1D-CNN only learns the temporal features of the audio, and does not consider the time structure and frequency features of the environmental sound. Therefore, most studies use two-dimensional features such as spectrograms as input to the model. Research [15] proposed a sound quality evaluation model combining Mel-frequency Cepstral Coefficient (MFCC) and CNN, and achieved a 95% recognition rate on the UrbanSound8k prediction set. Boddapati [16] used AlexNet and GoogleNet to evaluate the recognition accuracy of three audio feature maps of spectroscopy, MFCC, and cross-recursive map (CRP) in environmental sound. The author conducted experiments on multiple public datasets and found that in most cases, when using spectrograms, the highest classification accuracy can be obtained. However, due to the strong noise interference of environmental sound, the use of a single input feature causes the neural network to be unable to obtain sufficient audio information. Therefore, many studies suggest extracting multiple features for training.

In the second type of method, researchers use multiple features as input to the network. Zhu et al. [17] proposed an end-to-end network based on multi-scale convolution and two-phase method by using waveform-based features and spectrogram-based features, called WaveMsNet. On the ESC datasets ESC-10 and ESC-50, the classification accuracy of WaveMsNet reached 93.75% and 79.10%, respectively. WaveNet combines two different features. Many studies use two-stream networks to extract features [18, 19, 20]. Literature [19] proposed a TSCNN-DS model, using two sets of four-layer convolutional layer CNN network to calculate the feature set composed of log mel spectrum, chromaticity, spectral contrast, and tone, and then the two sets of networks are fully connected Layers to achieve integration. This model performs very well on the UrbanSound8K dataset. But the input of the model requires a combination of many features, which is too complicated.

In the third category of methods, research suggests expanding experimental data. Data expansion is a common method to improve the accuracy of ESC. Salamon [21] et al. conducted experiments on the UrbanSound8K dataset using four different audio data enhancement (distortion) techniques, and the accuracy rate obtained was 79%. Mushtaq [22] et al. proposed an offline data expansion method, using a deep convolutional neural network (DCNN), and the best accuracy rate obtained on the UrbansSound8K dataset was 95.37%. It is proved that the use of CNN technology combined with audio data enhancement can make the environmental sound classification obtain better generalization effect. But its accuracy rate needs to be improved. Literature [23] uses Generative Adversarial Network (GAN) to propose a new technology for audio data enhancement, which improves the classification accuracy of UrbanSound8K to 97.03%. However, this expansion method has two disadvantages. The first is to expand the data indiscriminately, without considering the impact of the imbalance of data types on accuracy. The second is to use a large increase in experimental data to improve the accuracy rate, and its training time is also significantly increased.

Most experiments show that combining feature extraction technology, data enhancement and CNN can indeed improve the accuracy of audio classification. However, in the existing research, the CNN model for ESC mostly adopts the traditional structure. To pursue the accuracy of classification, these networks usually use a deeper structure and add full connection layers at the end. Therefore, the parameters and calculation of neural network are large, and

the training speed is slow. The model using multiple streams leads to the doubling of the number of network parameters. To solve the above problems, many lightweight networks such as SqueezeNet [24], ShuffleNet [26], MobileNet [28], etc. have been proposed. These lightweight networks have been proved to obtain competitive accuracy in limited storage space. However, lightweight networks also have a loss in accuracy. In addition, the data enhancement methods in most studies have doubled the amount of data. They greatly increased the training time and did not study the characteristics of the dataset. Therefore, how to expand data efficiently is a problem worth discussing.

Combining the problems existing in environmental sound classification, this research has made two main contributions. First, to solve the problems of inconsistent data size and data imbalance in the dataset, we propose a data repair and dataset balancing strategy. It overcomes the defects of dataset and improves the stability of classification. Second, we propose a high-precision and low-parameter environmental sound classification model called TSLCNN-DS. This model designs two lightweight networks, MFCC-Net and GFCC-Net, which are stacked by different lightweight modules. These two networks use Mel-frequency Cepstral Coefficient (MFCC) and Gammatone Filter Cepstral Coefficients (GFCC) as input signals to capture sufficient audio information.Then, to fully combine the information extracted by the two networks, the D-S evidence theory is used to fuse the output results of the two CNNs. The experimental results show that the two networks proposed in this paper show strong recognition performance on the UrbanSound8K [27] dataset, which is significantly better than other methods. The fused model exceeded the classification accuracy achieved by the two single-network models, and the best accuracy rate on the UrbanSound8K dataset reached 98%.

Methods

Dataset preprocessing. *Data repair strategy.* UrbanSound8K is a public dataset currently used in the research of automatic urban environmental sound classification. The dataset contains 8732 audio clips (≤4s) with marked sound categories, and the sound categories cover ten categories. Figure 1 shows the number of audio clips contained in each sound category in the dataset.

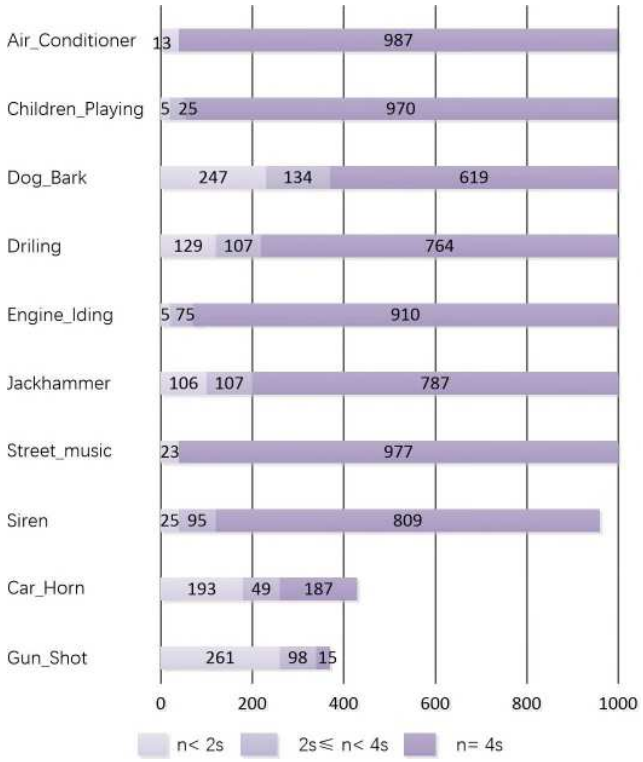

Figure 1. UrbanSound8K dataset (where the colors from light to dark respectively indicate that the audio length is less than 2s, greater than (including or equal to) 2 seconds but less than 4s, which is equal to 4s).

As shown in Figure 1, in UrbanSound8k, there are 1668 samples with a length of less than 4s, accounting for 19.1% of the total dataset. Among them, there are 981 audios whose length is less than 2s. Among them, there are

981 audios whose length is less than 2s. To unify the size of the input data, many studies performed zero-padded operations on data less than 4s in the preprocessing stage. For samples less than 2s, using the zero-padding method will miss many features, which is not conducive to classification. We proposed a new random patching method:

1. Suppose the sampling duration is n, where 0<n<2s. Then randomly select an audio from the same type, and extract a sample of 2-n seconds from it. In this way, the length of the audio is padded to 2s. Finally, copy the entire sample to get 4s audio.

2. Suppose the sampling duration is n, where 2s≤n≤4s. Then a data segment with a duration of 4-n seconds is randomly selected to make the audio reach 4s at a time.

Data balancing strategy. Through Figure 1 we find that in the dataset, there are seven categories of audio data with 1000 samples, and three categories with less than 1,000 samples. Particularly, there are less than 500 samples of car horns and gunshots, which is less than one-half of other categories. This makes the model face the problem of data imbalance in the process of classification. To solve this problem, we introduced audio rotation, time stretching, and pitch shift to increase the amount of training sample data. We set the total number of samples of a certain type of audio as N. For audio categories with less than 1000 samples, we need to add (1000-n) samples to make this type of data consistent with the number of samples in other categories. For categories with a sample size greater than 500, we randomly select (1000-N) existing samples. Then randomly extract one of the three methods of audio rotation, time stretching, and pitch shift to process the extracted samples. For categories with a sample size of less than 500, we divide the (1000-N) data into 3 equal parts. That is, randomly select [(1000-N)/3] samples from the original samples. After sampling 3 times, the three batches of samples are respectively subjected to time stretching, gene shift, and audio rotation operations, so that the number of samples reaches 1000 (the pseudo code is shown in Algorithm 1). It should be noted that for audio categories that have reached 1000 samples, we will no longer expand the number of samples. The small increase in experimental samples not only solves the problem of unbalanced data samples, but also does not double the amount of data to save the training time of the model.

Algorithm 1 Data balancing algorithm when the sample size is less than 500

Input: N: number of samples; name: sample name
freq: sampling frequency; list: signal list of samples
Output: *audio*: new audio samples generated by data processing;

```
ave ← (1000 – N)/3
List1 ← random.choice( ave: list);
List2 ← random.choice( ave: list);
List3 ← random.choice( (1000 – n – 2*ave): list);
for s init to limit by List_1 do
    output.write_wav(name + '_1' , pitch_shift(s, freq, n_step = 3))
end
for s init to limit by List_2 do
    output.write_wav(name + '_2' , time_stretch(s, freq, rate = 0.7))
end
for s init to limit by List_3 do
    output.write_wav(name + '_3' , roll(s, frep*2))
end
```

Model. TSLCNN-DS is a two-stream CNN that combines MFCC-Net and GFCC-Net. We use MFCC-Net to extract the MFCC features of the audio, which is a stack of M-Modules. M-Module is composed of parallel attention mechanism and Bottleneck [24]. At the same time, we use GFCC-Net to extract the GFCC features of the audio, which is a stack of G-Modules. G-Module is composed of feature multiplexing structure and Fire Module [29]. This section introduces the structure of the two networks and the D-S fusion algorithm in detail.

*M-Module based on parallel attention mechanism and Bottleneck***.** MobileNet V2 [24] is a network with Bottleneck as the basic module. The use of depth separable convolution and inversion residuals can reduce the number of network parameters. The structure of ordinary convolution and depth separable convolution is shown in Figure 2. The depth separable convolution splits ordinary convolution into two parts: the first layer uses channel-by-channel convolution to convolve different input channels separately; the second layer uses point-by-point convolution to combine the output of each channel.

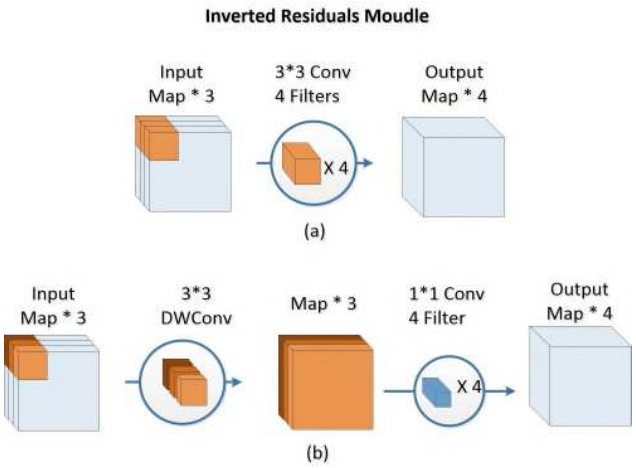

Figure 2. (a) Ordinary convolution (b) Depth separable convolution

The standard convolution module is shown in Figure 3(a), and the Bottleneck is shown in Figure 3(b). Bottleneck is a reversal residual structure. In the Bottleneck, the input first goes through a 1*1 point-by-point convolution + ReLU6 layer to expand the number of channels in the low-dimensional space, and the dimension is expanded from n to kn dimensions; then goes through 3*3 deep convolution + ReLU6 layer extracts features from the previous expansion layer; finally, the number of channels is compressed back through the 1*1 convolution kernel, and the dimension is reduced from kn to n dimension. Where k is the expansion multiple of the dimension, and the default is 6. The dashed part in Figure 3(b) indicates that when stride = 1, sum will be used to add the input and output; when stride = 2, there is no shortcut to connect the input and output features.

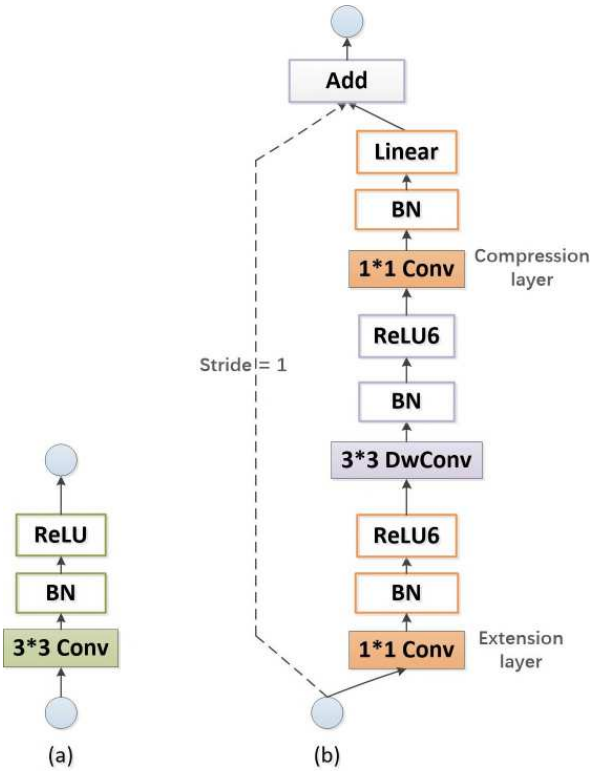

Figure 3. (a) Ordinary convolution module (b) Basic module of MobileNet V2

To further improve the accuracy of this module in environmental sound recognition, we add a parallel convolution block attention module (P-CBAM) to Bottleneck. Enable the lightweight network to filter out useless information and effectively improve performance. The improved M-Module structure is shown in Figure 4. When Stride = 1, use the input features of this module to generate attention features through the parallel attention module. Then the attention feature, the output feature of the inverted residual structure and the input feature are added together as the output feature of the next module.

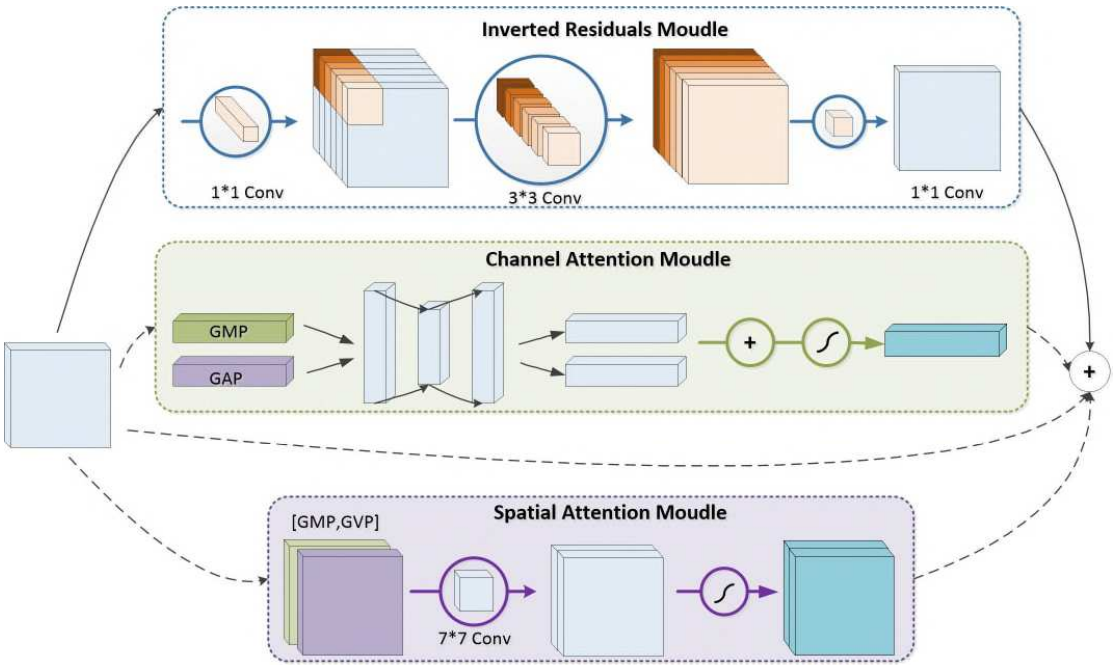

Figure 4. Bottleneck layer based on parallel attention block convolution module (M-Module)

The CBAM [29] module is divided into two sub-modules: channel and space. CBAM takes the input feature F to obtain the attention feature along the channel dimension, namely $M_c(F)$. The $M_c(F)$ and the feature F are multiplied element-wise to generate the input F_c required by the Spatial attention module. Then F_c obtains the spatial attention feature along the spatial dimension, namely $M_s(F)$. Finally, $M_s(F)$ and the input feature F are multiplied element-wise to obtain the final generated feature. The realization formula is:

$$\begin{cases} F_c = M_c(F) \otimes F \\ F_s = M_s(F_c) \otimes F_c \end{cases} \tag{1}$$

The information obtained from formula (1) is that the feature map input by the spatial dimension is the feature map F_c generated by the channel dimension, which is a serial structure. Since the features extracted from the channel dimension will directly affect the spatial dimension, the use of this structure will result in the loss of part of the information. Therefore, we changed this serial structure to a parallel structure. The two convolutional blocks can directly learn the input features, thereby obtaining P-CBAM. P-CBAM passes the input feature map through channel attention and spatial attention respectively to generate attention features. Finally, the two feature maps are weighted with the original input feature to obtain the output feature map. The process is the formula:

$$F_c = M_c(F) \otimes F \tag{2}$$

$$F_s = M_s(F) \otimes F \tag{3}$$

The parallel attention structure can effectively avoid the interference of the space attention module to the channel attention module. Thereby improving the accuracy of CBAM. In addition, since CBAM is a lightweight general-purpose module, the addition of P-CBAM did not significantly increase the parameters of MFCC-Net.

G-Module based on residual learning and Fire Module. In SqueezeNet, the key to parameter compression is to construct the Fire Module by compressing the size of the convolution kernel. Its basic structure is shown in Figure 5(a). The Fire module consists of two layers of convolution: the first layer is a Squeeze layer composed of 1 x 1 convolution, with S1 filters; the second layer is an Expand layer that combines 1 x 1 convolution and 3 x 3 convolution, and the number of filters is E1 and E3, respectively. To minimize the number of 3 x 3 input channels, the author of SqueezeNet suggested that the value of S1 should be smaller than the sum of E1 and E3. The entire SqueezeNet is built using Fire Module.

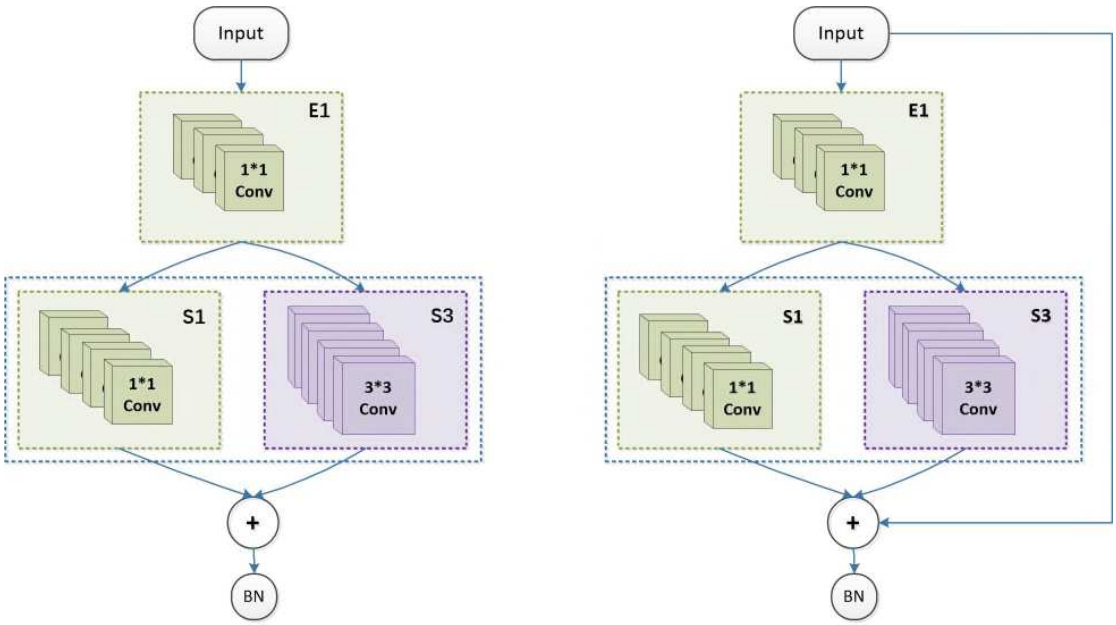

Figure 5. (a) Fire Module (b) Fire Module with residual learning (G-Module)

To strengthen the transfer of features, we have introduced residual learning in Fire Module. The improved module is called G-Module, and its basic structure is shown in Figure 5(b). We fuse the input features of this layer with the output features of the Expand layer as the input of the next layer. This improvement can make the network stacked by G-Module extract features more efficiently. There is also the effect of regularization, which can effectively suppress overfitting [30]. In addition, to control the number of convolution kernels in the compression layer and the expansion layer, we set the G-Module with S = 16 and E = 64 for experiments.

Information Fusion Based on D-S Evidence Theory. To better integrate the audio information extracted by MFCC-Net and GFCC-Net, we use D-S evidence theory (9) to fuse the output tensors of the two networks to generate classification results. D-S evidence theory is a mathematical theory proposed by Demspter and his student Shafer. This theory can deal with uncertain problems and is often used in a variety of information fusion. We use this theory to merge the "evidence" provided by the two networks to make decisions. The specific implementation method of D-S evidence theory is as follows:

①The recognition frame $X = \{x_1, x_2, \dots, x_n\}$ is the set of all possibilities in the uncertainty problem;

②The basic probability is the probability of occurrence of each hypothesis in the recognition framework.

③Basic Probability Assignment (BPA) is the process of calculating the basic probability of each "witness" for each situation in the entire X domain.

④The Mass function is a function used in the process of basic probability assignment, denoted as M(x). The Mass function must satisfy the following formula:

$$\begin{cases} 0 \le M(x) \le 1 \\ M(\emptyset) = 0 \\ \sum_{x \subseteq X} M(x) = 1 \end{cases} \tag{4}$$

⑤We use the output of MFCC-Net and GFCC-Net as the evidence source to combine the evidence, and set their Mass functions as $M_1(x)$, $M_2(x)$. For $\forall A \subseteq X$, the Dempster synthesis rule of $M_1(x)$, $M_2(x)$ is:

$$\begin{cases} M_{1\oplus M_2(A)} = \dfrac{1}{K}\sum\limits_{B\cap C=A} M_1(B)\cdot M_2(C) \\ M_{1\oplus M_2(\emptyset)} = 0 \\ K = \sum\limits_{B\cap C\neq\emptyset} M_1(B)\cdot M_2(C) \end{cases} \tag{5}$$

Among them, the output of $M_1 \oplus M_2(A)$ is the result of information fusion using D-S evidence theory.

TSLCNN-DS. We use a convolutional network composed of MFCC-Net and GFCC-NET to learn feature representations from MFCC and GFCC. Then use D-S evidence theory to fuse the output of the two networks to get the classification result. The two-stream model (TSLCNN-DS) proposed in this paper adopts a modular design method. Figure 6 shows the architecture of our model:

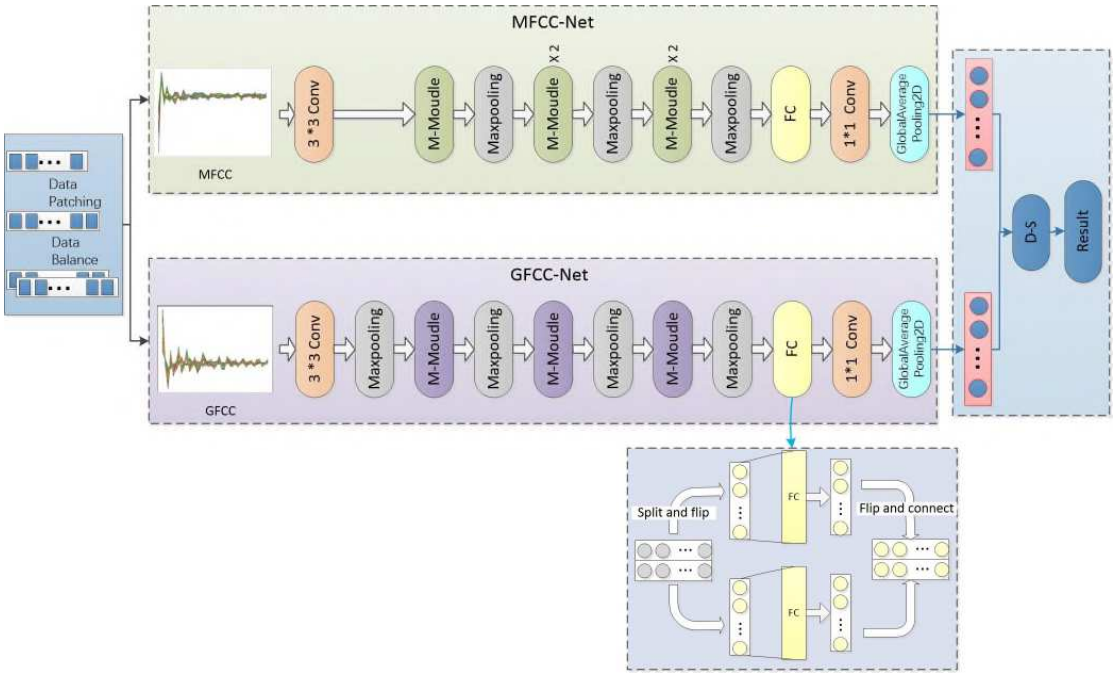

Figure 6. Model structure of TSLCNN-DS

It connects the stacking blocks composed of M-Module and G-Module to form the basic framework of the network. In the composition of MFCC-Net and GFCC-Net, we use 5 M-Modules and 3 G-Modules respectively, and insert multiple maximum pooling layers in the middle. After completing the convolution and pooling operations, the two networks respectively use a fully connected layer with 64 filters to connect the distributed features. Then put the output features into a 1×1 convolutional layer with 10 filters. Finally, after the global average pooling layer, the softmax function is used to complete the classification. The fully connected layer here is a fully connected calculation for each row vector of the output feature of the previous layer. Then concatenate the calculated vectors into a matrix. There are two advantages of using this fully connected layer. One is to achieve the effect of connecting local features. The second is to greatly reduce the number of parameters and calculations of the fully connected layer. It can achieve the best classification effect on the basis of a small increase in the amount of network parameters.

In the two-stream network, the first and second M-Modules of MFCC-Net use the same parameters, and the third and fourth M-Modules also use the same parameters. Therefore, in Figure 6, we only show three M-Modules, and mark "×2" at the top to indicate that this M-Module is repeated twice. In GFCC-Net, all three G-Modules use the same parameters. To reduce computational complexity, multiple maximum pooling layers have been added to our network. Table 2 shows the detailed configuration used for the experiment. Among them, Input_1 is the input shape of MFCC-Net, and Input_2 is the input shape of GFCC-Net.

Table 2. Network parameters of MFCC-Net (left) and GFCC-Net (right)

Input_1	Operator	k	c	n	stride	Input_2	Operator	S	E	c	stride
40*173	Conv2d(3*3)	-	64	1	2	40*169	Conv2d(3*3)	-	-	64	1
						40*169	Maxpooling(2*2)	-	-	-	2
20*87	M-Module	6	16	2	1	20*87	G-Module	16	64	-	1
20*87	Maxpooling(2*2)	-	-	-	2	20*87	Maxpooling(2*2)	-	-	-	2
10*43	M-Module	6	24	2	1	10*43	G-Module	16	64	-	1
10*43	Maxpooling(2*2)	-	-	-	2	10*43	Maxpooling(2*2)	-	-	-	2
5*21	M-Module	6	32	1	1	5*21	G-Module	16	64	-	1
5*21	Maxpooling(2*2)	-	-	-	2	5*21	Maxpooling(2*2)	-	-	-	2
2*10	Dense	-	6	1	-	2*10	Dense	-	-	64	1
2*10	Conv2D（1*1）		10	-	1	2*10	Conv2D（1*1）	-	-	10	1
2*10	GAP	-	-	-	-	2*10	GAP	-	-	-	-
10	Activation	-	-	-	-	10	Activation	-	-	-	-

Result

Experimental device. For training, since environmental sound recognition is a multi-classification problem, we choose the cross-entropy function as the loss function and Adam as the optimizer. Each batch randomly selects 32 feature maps from the training set, without repeating them. Then set the learning rate to 0.001 and train 80 times. To avoid experimental contingency, all our experiments use 5-fold cross-validation. All experiments are done in the python environment of version 3.6. Use Keras library and TensorFlow backend, and use Nvidia GT 720 GPU and 4 GB memory to train all the proposed networks.

To evaluate the performance of the mothed proposed in this study, we conducted experiments on the public dataset UrbanSound8k. This chapter discusses the experimental results and compares them with previous work.

Feature selection. Choosing the appropriate input data has a great influence on the recognition model. At present, the most commonly used feature of audio classification systems is Mel Frequency Cepstral Coefficient (MFCC). We use MFCC to solve the ESC problem. However, environmental sound has complex strong background noise. The noise immunity performance of a single MFCC feature is not strong. Therefore, we use Gammatone Filter Cepstral Coefficients (GFCCs) with strong anti-noise performance to supplement the deficiencies of the features extracted by the filter. This article uses a sampling rate of 22.05 KHz to sample all samples used in the experiment. We use the Librosa library to extract MFCC features. A window with a frame length of 2048 and a frame movement of 512 are used to extract features from the audio clip, and 40 MFCC coefficients are retained. The resulting feature matrix is 40×173. The specific extraction method of GFCC features is as follows:

(1) The voice signal is input through the 128-channel Gammatone filter bank;

(2) Take the absolute value of the filtering of each channel, which produces a time domain (T-F) representation;

(3) Take the logarithm of the filter after taking the absolute value;

(4) Use DCT to extract cepstrum features to reduce the correlation between features in various dimensions.

For GFCC, we take the features of the first 40 channels and get a 40×169 feature vector.

To achieve the best classification effect, different networks need to be combined with input features. We put the MFCC and GFCC features into the unimproved MobileNet V2 and Squeezenet respectively for comparison experiments (parameters are shown in Table 2, and the results are shown in Table 1). The experiment uses zero-

filled audio for UrbanSound8k. Table 2 shows that MFCC has a better classification effect on MobileNet V2, while GFCC has the better performance on SqueezeNet. Since then, we have used MFCC to improve the experiment of MobileNet V2, and use GFCC as the input features to improved Squeezenet.

Table 1. Comparison of the combination of different features and the network

Feature	MobileNet V2	SqueezeNet
MFCC	94.4%	90.1%
GFCC	89.0%	91.5%

Comparison of data repair and data equalization with the original dataset. We evaluated the proposed network on datasets of zero padding, data patching, and data balancing. Table 3 shows the total duration of the three datasets. Table 3 shows that data-patching dataset adds about 1 hour of audio length to the zero-padding dataset, while data-balancing dataset adds 2.4 hours of audio length to the zero-padding dataset. The experimental results are shown in Table 4. The accuracy evaluation includes precision, recall, F1-score, the highest accuracy rate and the lowest accuracy rate. Among them, precision and recall represent the classification accuracy of the model. The F1-score and the difference between the highest and lowest accuracy represent the stability of the model.

Table 4 shows that both data patching and data balancing strategies improve the recognition rate of audio classification. The accuracy of MFCC-Net on the data-patching dataset has increased by 1.3%, and the accuracy of data-balancing has increased by 2%. On the three datasets, the difference between the highest accuracy rates and lowest accuracy rates is about 2%. MFCC-Net achieved the highest F1-score on the data-balancing dataset. At the same time, the accuracy of GFCC-Net in data-patching has increased by 1.8%, and the accuracy of data-balancing has increased by 3.9%. On the zero-padding dataset and the data-patching dataset, the difference between the highest and lowest accuracy is about 2.5%. On the balancing dataset, the difference between the highest and lowest accuracy is only 0.6%. F1-score increased by 4.1%.

Combining Table 3 and Table 4, we find that data patching and data balance add 2.4 hours of audio length to the dataset. Improved the recognition accuracy of GFCC on GFCC-Net by 4.14%. At the same time, the recognition accuracy of MFCC on MFCC-Net has been improved by 2%. Among them, the highest accuracy rate of MFCC-Net reached 95.86% and 97.5% on the dataset of data-patching and data-balancing. From the above data, we can conclude that data-patching and data-balancing strategies are effective. The accuracy and stability of the network model can be improved. Combining the total duration of the audio with the accuracy of the model makes the method proposed in this article a very competitive advantage.

Table 3. The total duration of the three datasets

	Zero-padding	Data-patching	Data-balancing
Total length (Hour)	8.75	9.70	11.12

Table 4 Compare the performance of zero padding, data patching and data balancing in different models

Model	Feature	Result	Zero-padding	Data-patching	Data-balancing
MFCC-Net	MFCC	Precision	94.50%	95.86%	96.48%
		Recall	94.52%	95.70%	96.46%
		F1-score	94.26%	95.68%	96.48%
		Best	95.80%	96.80%	97.50%
		Worst	92.30%	95.10%	95.60%
GFCC-Net	GFCC	Precision	91.54%	93.34%	95.40%
		Recall	91.30%	92.94%	95.34%
		F1-score	91.20%	93.06%	95.34%
		Best	92.90%	94.60%	95.80%
		Worst	90.40%	92.10%	95.20%

Experimental results of MFCC-Net and GFCC-Net. In order to compare the advanced nature of MFCC-

Net and GFCC-Net, we added different structures to the bottleneck and Fire module for experiments. Table 5 shows the experimental results. We use two unimproved basic modules as baselines. Among them, the method of combining RNN is to use 2 RNNs (GRU) and 32 recursive units after the last stacked block of the network, and the purpose is to summarize the time pattern at the top of the two-dimensional CNN. The assumption of this model is that RNN can aggregate temporal patterns better than CNN. But obviously this has no effect on our network. To add multi-scale convolution to the bottleneck is to add 5×5 and 7×7 depth separable convolution modules to the separable convolution part of the inverse residual network. The purpose is to aggregate features extracted from different receptive fields. In addition, using connections to connect input and output functions in a GFCC network will increase the number of network parameters. In order to explore the influence of the parameter amount on the stacking block network of the fire protection module, we set the network parameters of the fire protection module to S=24 and E=96. The number of parameters corresponding to the stacked block network has also increased.

Table 5 shows that, compared with the unimproved bottleneck, the structure of the combined RNN does not improve the classification accuracy of the model. After adding multi-scale convolution and CBAM, the accuracy of the bottleneck is increased by 0.76% and 0.82%, respectively. After adding the Parallel Convolutional Block Attention Module (P-CBAM), the accuracy of the M-module stacked block network is increased to 96.48%, which is 1.12% higher than the unimproved bottleneck. At the same time, adding RNN and CBAM to the Fire module increased the parameters of the stacked block network, but the classification accuracy did not increase, but decreased. With the addition of residual learning, the recognition rate of the G-module superposition block network increased from 93.60% to 95.40%, an overall increase of 1.8%. The method of setting the network parameters to S=24 and E=96 has a 0.86% lower accuracy than the residual learning method, and its parameters are also greatly increased. At the same time, adding RNN and CBAM to the Fire module increased the parameters of the stacked block network, but the classification accuracy did not increase, but decreased. With the addition of residual learning, the recognition rate of the G-module superposition block network has been greatly improved, from 93.60% to 95.40%, an overall increase of 1.8%. We see that after setting the network parameters to S=24 and E=96, the accuracy of the network is indeed improved. But compared with the residual learning method, its accuracy is reduced by 0.86%, and its parameters are also greatly increased. Therefore, adding a residual structure to the Fire module can effectively improve the classification performance.

Table 5. Improved experimental results on two modules

Model	Param	Accuracy
Bottleneck	37642	95.36%
Bottleneck + RNNs	50762	95.40%
Bottleneck + Multiscale convolution	76618	96.12%
Bottleneck + CBAM	51350	96.18%
Bottleneck + P-CBAM (M-Module)	51350	**96.48%**
Fire Module	45818	93.88%
Fire Module + RNN	71226	93.60%
Fire Module + CBAM	67336	90.93%
Fire Module (Set S = 24, E = 96)	94162	94.54%
Fire Module + Residual learning (G-Module)	70396	**95.40%**

Figure 7 shows the typical training curves of MFCC_Net, MobileNet, GFCC_Net, and SqueezeNet. According to the classification accuracy after 80 training iterations, GFCC_Net, MFCC_Net and MobileNet have similar accuracy rates. Combining the network parameters and training accuracy of GFCC_Net, we can again conclude that the improvement of this research on SqueezeNet has achieved great results.

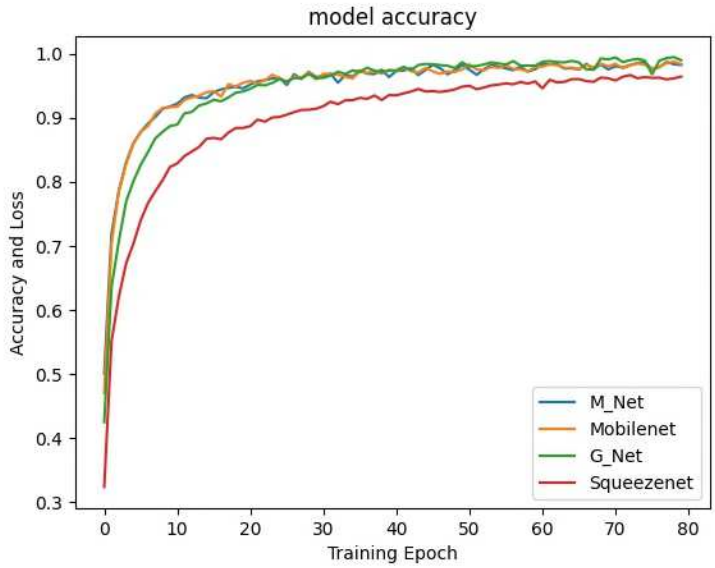

Figure 7. The accuracy training curves of MFCC_Net (M_Net), MobileNet (Mobilenet), GFCC_Net (G_Net), and SqueezeNet (Squeezenet) on UrbanSound8k balancing dataset

Comparison of information fusion methods. To prove that the method of using D-S evidence theory at the decision-making level is effective, we compared the results obtained by different feature fusion methods. Table 6 shows the impact of different feature fusion methods on classification accuracy. Among them, the method shown by TSLCNN is to perform fusion on the feature layer. This method uses the Concat operation after the fully connected layers of the two networks to fuse the features output by the two CNNs. Then put the features into a 1×1 convolutional layer with 10 filters. Finally, after global average pooling, the softmax function is used to complete the classification. TSLCNN-Average is also a fusion at the decision-making level. Different from the D-S method, TSLCNN-Average directly averages the classification results obtained by the two networks.

Table 6. Experimental results of different fusion algorithms

Class	TSLCNN	TSLCNN-Average	TSLCNN-DS
AC	98.00%	98.60%	99.00%
CH	97.20%	98.60%	98.80%
CP	93.40%	94.20%	95.80%
DB	96.20%	96.80%	97.00%
Dr	94.60%	96.40%	96.40%
EI	99.40%	97.20%	99.40%
GS	99.60%	98.80%	99.20%
JH	96.40%	96.80%	96.80%
Si	95.60%	95.40%	95.60%
SM	96.60%	95.00%	96.40%
Average	96.70%	96.78%	97.44%

Table 6 shows that in most cases, the method of fusion at the decision-making level using D-S evidence theory has higher accuracy. Among them, the accuracy of the D-S fusion method is about 0.7% higher than the average classification result. The average fusion method is not much improved compared to the fusion method on the feature layer. According to the analysis, when fusion is performed on the feature layer, the size and number of feature maps of the two networks are equal. It is worth noting that the average accuracy of the TSLCNN-DS method on each classification is higher than 95%. This further proves the superiority of fusion model using D-S decision analysis method. The typical confusion matrix of TSLCNN-DS on the UrbanSound8k balancing dataset is shown in Figure 8.

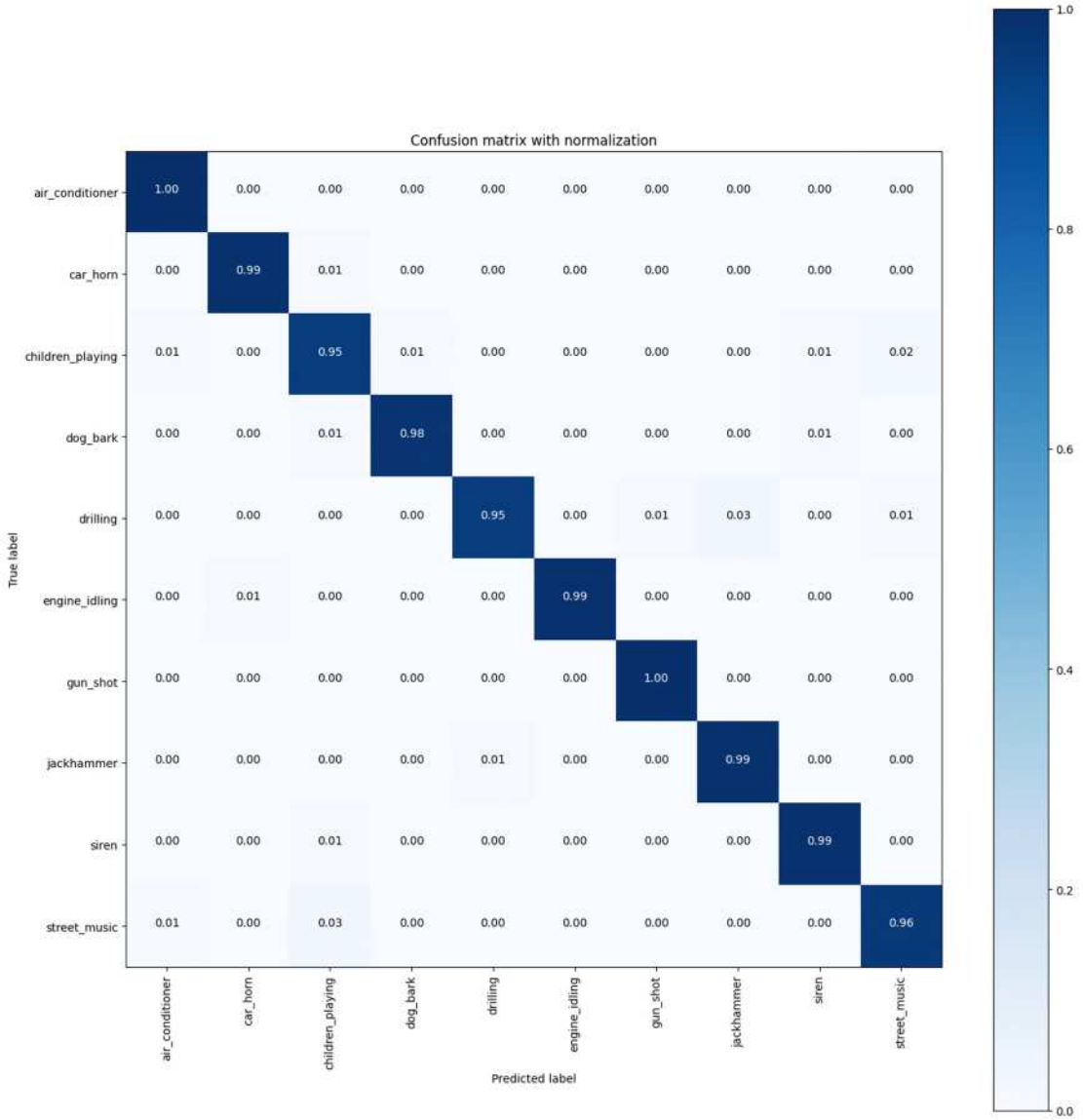

Figure 8. Confusion matrix of TSLCNN-DS on the UrbanSound8k balancing dataset

Discussion. To further verify the effectiveness of the model, we compared it with other models experimented on the UrbanSound8K dataset. From the results shown in Table 7, the accuracy of MFCC_Net using MFCC features is as high as 96.48%, which is 0.78% higher than TSCNN [20] which uses RAW and LogMel and uses a two-network model. The performance of GFCC_Net is also better than most network models, such as DCNN[22] using LogMel as input and 1D CNN Gamma [32] using raw signals as input, and GoogleLeNet [16] using combined features. Both single network models use parameters less than 0.1M to obtain excellent classification results.

In addition, we compare the total length of audio used for training in Table 7. Compared with DCNN [22], which expanded the dataset by 6 times, we only added 2 hours of data. While saving model training time, the accuracy rate is increased by 2.1%. At the same time, we spent an hour more data than TSCNN-DS [19], and the accuracy rate increased by 1.7%. These data once again prove that the data patching and data balancing strategy is cost-effective.

PiczakCNN [14] is often used as the baseline for deep learning methods for ESC tasks. It can be seen from Table 7 that the total parameters of TSLCNN-DS are significantly lower than PiczakCNN, while TSLCNN-DS is better than PiczakCNN in classification accuracy (24.7% higher). Compared with Dong [20] and others who use the two-stream model, our accuracy is improved by 1.4%. Compared with Li [18] et al. who use the same fusion algorithm, our method has made great progress (5.2% higher). More significant is that TSLCNN-DS achieved an average accuracy rate of 97.44%, and its highest accuracy rate reached 98%, but only used 0.12M parameters. Other network

models such as GoogleLeNet [16], DCNN[22], etc., use several dozen times the parameters of TSLCNN-DS, but the accuracy is far less than TSLCNN-DS. TSCNN-DS achieved a lower accuracy rate of TSLCNN-DS, but it took 15.9 M parameters. It is one hundred times the parameters of TSLCNN-DS.

Table 7 Comparison with the experimental results of existing studies

Model	Total length (Hour)	Feature	Param	Accuracy
PiczakCNN [14]	8.75	LogMel	31.53 M	72.70%
DCNN [22]	67.9	LogMel	3.17 M	95.30%
1D CNN Gamma [32]	8.75	Raw	0.55 M	89%
DS-CNN [18]	8.75	Raw + LogMel	-	92.20%
GoogleLeNet [16]	8.75	Spec + MFCC + CRP	6.7 M	93.00%
TSCNN [20]	8.75	Raw + LogMel	-	95.70%
TSCNN-DS [19]	9.70	Mel-CST-LM	15.9 M	97.20%
GFCC-Net	11.12	GFCC	0.07 M	**95.40%**
MFCC-Net	11.12	MFCC	0.05 M	**96.48%**
TSLCNN-DS	11.12	MFCC + GFCC	0.12 M	**97.44%**
TSLCNN-DS(Best)	11.12	MFCC + GFCC	0.12 M	**98.00%**

Conclusion

This paper proposes a two-stream model based on a lightweight convolutional neural network for environmental sound classification. We first proposed two modified lightweight convolution modules, which greatly improved the classification accuracy of the network formed by stacking these two modules. Two-stream CNN uses MFCC and GFCC matrices as input data respectively. In this way, sufficient environmental audio event information can be fully extracted. Then, through the method of information fusion, the feature information extracted by the two stacked block networks is fused to obtain a higher accuracy rate. Specifically, using data patching and balance strategies, a parallel attention mechanism was added to Bottleneck, which increased the accuracy of MFCC by 2%; residual learning was added to Fire Module, which increased the accuracy of GFCC by 4.1%. Then use D-S evidence theory to fuse the information output by the two networks. Experiments show that the use of D-S evidence theory for fusion at the decision-making level can effectively fuse the output information of the two networks, and significantly break through the performance Bottleneck of a single network. In addition, we have added data repair and equalization strategies to the data preprocessing part. The increase of experimental data improves the performance of model classification, and the balance of data samples can improve stability. Finally, the experimental results show that we use only 0.12 M parameters to achieve an average recognition rate of 97.44% and the highest accuracy rate of 98% on the UrbanSound8K dataset.

Acknowledgements

We thank the key R&D projects of Shandong Province (No.2020JMRH0201) and the key R&D projects of Shandong Province (No.2019JMRH0109) for their support.

Author Contributions

All authors conceived and designed the study. J. F. carried out the experiments. J. F., B. Y. and Z. D. analyzed the experimental results. All the authors wrote manuscripts. X. J. revised the manuscript. All authors have approved the submitted version.

Competing interests

The authors declare no competing interests.

References

1. Weninger, F. & Schuller, B. Audio recognition in the wild: Static and dynamic classification on a real-world database of animal vocalizations. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 337-340 (2011)
2. Lokesh, S. & Devi, M. R. Speech recognition system using enhanced mel frequency cepstral coefficient with windowing and framing method. *Cluster. Comput* **22**, 11669–11679 (2019)
3. Jakubec, M. & Chmulik, M. Automatic music genre recognition for in-car infotainment. *Transportation Research Procedia* **40**, 1364-71 (2019)
4. Meghana, M. et al. Hand gesture recognition and voice controlled robot. *Materials Today: Proceedings* **33**, 4121-4123 (2020).
5. Kaddoura, T. et al. Acoustic diagnosis of pulmonary hypertension: automated speech- recognition-inspired classification algorithm outperforms physicians. *Scientific Reports* **6**, 33182 (2016).
6. Theodorou, T., Mporas, I. & Fakotakis, N. Automatic sound recognition of urban environment events. In *International Conference on Speech and Computer* 129-136 (2015).
7. Khunarsal, P., Lursinsap, C. & Raicharoen, T. Very short time environmental sound classification based on spectrogram pattern matching. *Information Sciences* **243**, 57-74 (2013)
8. Tokozume, Y. & Harada, T. Learning environmental sounds with end-to-end convolutional neural network.In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 2721-2725 (2017)
9. Chen, Y., Guo, Q., Liang, X., Wang, J. & Qian, Y. Environmental sound classification with dilated convolutions. *Applied Acoustics* **148**, 123-32 (2019)
10. Al-Hattab, Y. A., Zaki, H. F. & Shafie, A.A. Rethinking environmental sound classification using convolutional neural networks: optimized parameter tuning of single feature extraction. *Neural Comput Appl* 1-12 (2021)
11. Tran, V. T. & Tsai, W. H. Acoustic-based emergency vehicle detection using convolutional neural networks. *IEEE Access* **8**, 75702-75713 (2020).
12. Dai, W., Dai, C., Qu, S., Li, J. & Das, S. Very deep convolutional neural networks for raw waveforms. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 421-425 (2017).
13. Chandrakala, S. & Jayalakshmi, S. L. Generative model driven representation learning in a hybrid framework for environmental audio scene and sound event recognition.In *IEEE Transactions on Multimedia* **22(1)**, 3-14 (2019)
14. Piczak, K. J. Environmental sound classification with convolutional neural networks. *In IEEE 25th international workshop on machine learning for signal processing (MLSP)* 1-6 (2015).
15. Jin, S., Wang, X., Du, L. & He, D. Evaluation and modeling of automotive transmission whine noise quality based on MFCC and CNN. *Applied Acoustics* **172**, 107562 (2021)
16. Boddapati, V., Petef, A., Rasmusson, J. & Lundberg, L. Classifying environmental sounds using image recognition networks. Procedia computer science. **112**, 2048-2056(2017)
17. Zhu, B., Wang, C., Liu, F., Lei, J. & Lu, Z. Peng, Y. Learning Environmental Sounds with Multi-scale Convolutional Neural Network. IEEE Access. In *2018 International Joint Conference on Neural Networks (IJCNN)* 1-8 (2018).
18. Li, S., Yao, Y., Hu, J., Liu, G., Yao, X. & Hu, J. An Ensemble Stacked Convolutional Neural Network Model for Environmental Event Sound Recognition. *Appl. Sci* **8(7)**, 1152 (2018).
19. Su, Y., Zhang, K., Wang, J. & Madani, K. Environment Sound Classification Using a Two-Stream CNN Based on Decision-Level Fusion. *Sensors* **19(7)**, 1733 (2019).
20. Dong, X., Yin, B., Cong, Y., Du, Z., & Huang, X. Environment sound event classification with a two-stream convolutional neural network. *IEEE Access* **8**, 125714-125721 (2020).
21. Salamon, J. & Bello, J. P. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal processing letters* **24(3)**, 279-283 (2017).
22. Mushtaq, Z. & Su, S. F. Environmental sound classification using a regularized deep convolutional neural network with data augmentation. *Applied Acoustics* **167**, 107389 (2020).
23. Madhu, A. & Kumaraswamy, S. (2019, September). Data augmentation using generative adversarial network for environmental sound classification. In *2019 27th European Signal Processing Conference (EUSIPCO)* 1-5 (2019).
24. Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J.: Keutzer, K.J.a.p.a. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *Computer Vision Pattern Recogni* (2016)
25. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. & Chen, L. C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* 4510-4520 (2018).
26. Zhang, X., Zhou, X., Lin, M. & Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition* 6848-6856 (2018)

27. Salamon, J., Jacoby, C. & Bello, J. P. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia* 1041-1044 (2014).
28. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. & Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *Computer Vision Pattern Recognit* (2017).
29. Woo, S., Park, J., Lee, J. Y. & Kweon, I. S. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)* 3-19 (2018)
30. Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K.Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* 4700-4708 (2017).
31. Shafer, G.: A mathematical theory of evidence: Princeton university press(1976).
32. Abdoli, S., Cardinal, P. & Koerich, A. L. End-to-end environmental sound classification using a 1D convolutional neural network. *Expert Systems with Applications* **136**, 252-263 (2019).