




# Squeeze-and-Excitation Self-Attention Mechanism Enhanced Digital Audio Source Recognition Based on Transfer Learning

Chunyan Zeng<sup>1</sup> · Yuhao Zhao<sup>1</sup> · Zhifeng Wang<sup>2</sup>  · Kun Li<sup>1</sup> · Xiangkui Wan<sup>1</sup> · Min Liu<sup>1</sup>

Received: 4 July 2023 / Revised: 22 August 2024 / Accepted: 24 August 2024

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

## Abstract

Recent advances in digital audio source recognition, particularly within judicial forensics and intellectual property rights domains, have been significantly propelled by deep learning technologies. As these methods evolve, they introduce novel models and enhance processing capabilities crucial for audio source recognition research. Despite these advancements, the limited availability of high-quality labeled samples and the labor-intensive nature of data labeling remain substantial challenges. This paper addresses these challenges by exploring the efficacy of self-attention mechanisms, specifically through a novel neural network that integrates the Squeeze-and-Excitation (SE) self-attention mechanism for identifying recording devices. Our study not only demonstrates a relative improvement of approximately 1.5% in all four evaluation metrics over traditional convolutional neural networks but also compares the performance across two public datasets. Furthermore, we delve into the self-attention

---

✉ Zhifeng Wang  
zfwang@cnu.edu.cn

Chunyan Zeng  
cyzeng@hbut.edu.cn

Yuhao Zhao  
102210253@hbut.edu.cn

Kun Li  
102210257@hbut.edu.cn

Xiangkui Wan  
xkwan@hbut.edu.cn

Min Liu  
liu\_min@hbut.edu.cn

<sup>1</sup> Hubei Key Laboratory for High-efficiency Utilization of Solar Energy and Operation Control of Energy Storage System, Hubei University of Technology, Nanli Road, Wuhan 430068, China

<sup>2</sup> Department of Digital Media Technology, Central China Normal University, Luoyu Road, Wuhan 430079, China

mechanism's adaptability across different network architectures by embedding the Squeeze-and-Excitation mechanism within both residual and conventional convolutional network frameworks. Through ablation studies and comparative analyses, we reveal that the impact of self-attention mechanisms varies significantly with the underlying network architecture. Additionally, employing a transfer learning strategy has allowed us to leverage data from a baseline network with extensive samples, applying it to a smaller dataset to successfully identify 141 devices. This approach resulted in performance enhancements ranging from 4% to 7% across various metrics, highlighting the transfer learning method's role in advancing digital audio source identification research. These findings not only validate the Squeeze-and-Excitation self-attention mechanism's effectiveness in audio source recognition but also illustrate the broader applicability and benefits of incorporating advanced learning strategies in overcoming data scarcity and enhancing model adaptability.

**Keywords** Digital audio forensics · Deep learning · Self-attention mechanism · Transfer learning · Few-shot learning

## 1 Introduction

In an era of rapid technological development, the use of digital media files has become intertwined with public life, with digital audio comprising a significant portion of these files [6]. While these data have made people's lives more convenient, they also pose various hidden risks [35, 36, 41–43]. For instance, in voice recognition systems, unauthorized individuals can exploit voice simulation software to mimic registered voices and thereby steal identity information [40, 44, 47]. Furthermore, accurate digital audio source recognition is crucial for forensic analysis and related tasks within the realm of justice [1, 37]. Consequently, research in this area holds considerable significance [38].

Inspired by advancements in speaker recognition technology [3, 9, 19, 26], digital audio source identification has also made significant progress [39, 45, 46]. Research on digital audio source identification comprises three essential components: preprocessing, feature extraction, and representation modeling. Preprocessing of digital audio, as the initial step in recognition, is necessary to address environmental noise that often interferes with the recording process. By undertaking relevant preprocessing, the audio becomes better suited for subsequent feature extraction work [18, 32].

Feature extraction of preprocessed audio represents another crucial aspect of digital audio source recognition. Buchholz et al. [5] proposed using frequency domain features obtained through Short Time Fourier Transform (STFT) as features for digital audio source forensics. Subsequent studies introduced unsupervised Random Spectral Features (RSF) [28], supervised Labeled Spectral Features (LSF) [29], and Sketches of Spectral Features (SSF) [21] features. While these features showed promise in experiments, they suffered from high redundancy and computational complexity. Inverse spectral coefficient-based characterization features such as Mel-Frequency Cepstral Coefficients(MFCC) [24], Linear Prediction Coding(LPC) [2], Linear Prediction Cepstral Coefficients(LPCC) [14], and Perceptual Linear Predictive(PLP) [16] gained

popularity and have since become the most effective and commonly employed representational information in digital audio source recognition studies.

Representation modeling, which involves generating algorithms that accurately match the extracted features, is also a crucial task in digital audio source recognition. Prior to the advent of machine learning, Gaussian Mixture Models (GMM) dominated the field. GMMs are probabilistic models that combine weighted and mixed Gaussian distributions of individual data to calculate posterior probabilities for making predictions. Consequently, feature extraction based on GMM models, such as Gaussian Super Vector (GSV) features [20] and i-vector features [10], emerged. The use of i-vector features significantly influenced subsequent deep learning research. However, while probability density modeling increased error tolerance in sample recognition, the results in open-set recognition remained poor.

With the emergence of deep learning, modeling and decision-making approaches have matured over time. The utilization of d-vectors through Deep Neural Network (DNN) network extraction [33] provided a better solution for efficient device information. Subsequently, an x-vector framework based on DNN embedding [33] was introduced to enhance model representativeness. Additionally, the advent of residual networks [15] enabled the stacking of deeper networks, which has proven pivotal in numerous deep learning-based studies.

Building upon the aforementioned research, this paper focuses on deep learning and introduces SE self-attentive mechanism within the convolutional network structure. Unlike traditional convolutional networks, our approach effectively incorporates both preceding and subsequent speech information at the frame-level, resulting in improved robustness. The main contributions of this paper are as follows:

- **Integration of Self-Attention Mechanisms** We systematically integrate self-attention mechanisms within both convolutional and residual network structures. This integration aids in examining the factors that influence the effectiveness of self-attention. By incorporating these mechanisms, our models gain the ability to process information more deeply and comprehensively. This reduces data redundancy and significantly improves both the efficiency and robustness of the model, which is crucial for handling complex audio data.
- **Advancements in Few-Shot Learning** Addressing the challenges of few-shot learning, we employ a transfer learning strategy. We initially pre-train our model using a dataset with ample similar data samples and subsequently fine-tune it on a smaller, few-shot dataset. This approach not only expedites the training process but also enhances the model's performance on limited data. Experimental results validate the substantial impact of transfer learning in improving the efficacy and adaptability of the training process under constrained data conditions.

The remainder of the paper is structured as follows: In Sect. 2, we present classical research approaches in the field based on our survey findings. Section 3 provides the mathematical notation and problem formulation associated with the theory. Detailed explanations of the deep learning methodologies employed in this study are presented in Sect. 4. The experimental design, measurement of the proposed approach, and analysis of the results are described in Sect. 5. Finally, Sect. 6 summarizes the conclusions drawn from this study, along with its limitations.

## 2 Related Work

This section provides an overview of different types of speech recognition models based on various construction methods. The models are classified into three categories: Gaussian mixture model-based digital audio source recognition models, support vector machine-based digital audio source recognition models, and deep neural network-based digital audio source recognition models.

### 2.1 Gaussian Mixture Model-Based Digital Audio Source Recognition

GMMs are used when the data is complex and cannot be accurately represented by a single Gaussian distribution. In GMMs, multiple Gaussian models are mixed with certain weights to form a probabilistic model that accurately represents the attribute features of the data. The GSV [20] is a feature data extracted from the mean vector of the GMM. The mean vector is a crucial component of the GMM model and has different representations. Therefore, the identification of GSV enables the recognition of the data used to build the GMM model.

Previous studies have used GMMs trained with the maximum likelihood function, but Kotropoulos et al. [22] and Zou et al. [48–50] trained GMMs as Universal Background Models (UBM) using the MFCC feature. They fine-tuned the UBM baseline system using the Maximum A Posteriori (MAP) algorithm to obtain independent GMM models. They achieved high recognition accuracy, with the former achieving 97.6% recognition accuracy using an Radial Basis Function-Neural Network (RBF-NN) classifier on the MOBIPHONE dataset, and the latter achieving an error rate of 2.08% for 14 mobile devices.

To address the limitations of training data length, Hanilci et al. [12] proposed using maximum mutual information to measure the Gaussian hybrid model. Comparative experiments demonstrated that training the hybrid Gaussian model using maximum mutual information was more effective than traditional training approaches, especially for short data.

### 2.2 Support Vector Machine-Based Digital Audio Source Recognition

Support Vector Machine (SVM) is generalized classifiers used for binary classification in supervised learning. SVM is originally designed for binary classification problems, so multiple classifiers need to be built for multi-classification problems. Two common approaches are “one-to-many” classifier construction and “one-to-one” classifier construction. In “one-to-many” classifier construction, samples of a certain category are grouped together during training, while the remaining samples are grouped into another category. K-SVMs are constructed for K categories, and the unknown category of samples is identified based on the largest classification function value.

Campbell [7] proposed a kernel function based on generalized linear judgments and the associated Mean Square Error (MSE) training criterion, achieving an equal error rate of 3.2% on the 1998 NIST speaker recognition evaluation task dataset. However, as the number of device classes increases, the computational cost of SVM classifiers

also increases exponentially, making SVM less feasible for a large number of device types.

### 2.3 Deep Neural Network-Based Digital Audio Source Recognition

The traditional GMM-UBM model is sensitive to channel noise. To address this issue, Dehak et al. [10] proposed reducing high-dimensional GSV to a low-dimensional vector called i-vector. The GMM/i-vector system effectively eliminates internal and channel variations of the speaker, leading to significant performance improvements. Inspired by the widespread use of deep learning, Lei et al. [23] introduced a DNN-based i-vector framework, which employs a DNN acoustic model to generate posterior probabilities instead of a GMM model. The resulting iso-error rate on the 2012 NIST speaker recognition evaluation task dataset was reduced by 30% compared to the GMM-UBM/i-vector based approach. Subsequent research in the field of deep embedding introduced the d-vector and x-vector frameworks, which have become seminal works in this area.

The d-vector framework utilizes the real identity of the speaker as the label during the training phase. In the testing phase, the d-vector represents the feature embedding of each frame by taking the output of the last hidden layer of the DNN as the frame's feature embedding. The average of the frame-level feature embeddings represents the audio's feature embedding. The x-vector framework is an extension of the d-vector framework, incorporating pooling operations to fuse frame-level speech signal features into speech-level signal features. It extracts frame-level signal features using a time-delay layer, combines the mean and standard deviation of these features through a statistical pooling layer to obtain speech-level signal features, and performs classification using a standard feedforward network.

Variani et al. [33] found that using the d-vector framework alone produced higher equal error rates than using the i-vector framework alone. However, when both frameworks were combined, it resulted in improved performance, achieving a 14% and 25% iso-error rate reduction over the baseline system. Snyder et al. [31] demonstrated that augmenting the data appropriately led to a 44% equal error rate reduction over the baseline system on the SITW Core dataset using the noisy corpus case.

DNN-based acoustic models exhibit superior performance in speech content-related recognition. These models not only provide more accurate frame-level recognition but also possess frame alignment capabilities, which are particularly advantageous in text-based digital audio source recognition. Geng et al. [11] demonstrated an 8.63% relative decrease in Word Error Rate (WER) compared to the baseline i-vector and x-vector systems by incorporating novel depth features and adaptively adaptive hybrid DNN/Time Delay Neural Network (TDNN) networks. Chakroun et al. [8] improved the i-vector-Probabilistic Linear Discriminant Analysis (PLDA) system by proposing a novel deep neural network based on Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN), achieving a recognition accuracy improvement of approximately 10%. It is worth noting that the computational complexity of DNNs is higher due to their multilevel network structure and the requirement for a larger number of labeled training samples, which should be taken into consideration.

**Table 1** Description of symbols used in this paper

Symbols	Descriptions
$X, Y, Z, U$	Represents the overall feature layer
$x_n, y_n, z_n$	Represents the $n$ th channel in a layer
$FL$	Integral convolution kernel
$fl_n$	Vector of the $n$ th dimension in the overall convolution kernel
$\partial()$	Sigmoid activation function
$F_1()$	Convolutional transform
$F_{gp}()$	Global average pooling operation
$F_{fc}()$	Through the fully connected layer
$W_n()$	Overall matrix of weighted values for the $n$ th fully connected layer
$W_{i,j}()$	The $j$ th vector in the $i$ th fully connected layer weighting matrix
$S_A$	Space exists for pre-trained models for transfer learning
$T_A$	The space that exists in the model obtained after fine-tuning transfer learning
$D_{SA}$	Data space for transfer learning to pre-train models
$D_{TA}$	The data space used for transfer learning to train the target network
$F_{SA}(\cdot)$	Source domain prediction function
$L(\cdot)$	Loss function

**Summary** Gaussian mixture model-based, support vector machine-based, and deep neural network-based models have been explored for digital audio source recognition. GMM-based models provide accurate representations of complex data, while SVM-based models offer good performance for a relatively small number of device classes. Deep neural network-based models, including the d-vector and x-vector frameworks, have demonstrated significant improvements in recognition accuracy, particularly in the presence of channel noise. These models leverage the power of deep learning and provide more accurate frame-level recognition and frame alignment capabilities, but at the cost of increased computational complexity and the need for abundant labeled training data.

### 3 Preliminaries

In this section, we provide an overview of the problem of digital audio source identification and introduce relevant definitions. Table 1 describes important symbols used throughout this paper for better understanding the proposed method.

**Definition 1: Digital Audio Source Recognition Tasks** The problem of digital audio source identification involves determining the specific recording device used to capture the digital audio in the input model. We can identify the device from the registered database  $\{X_a^e \mid a = 1, 2, \dots, A\}$ . This problem can be formulated as follows:

$$a^* = \underset{a}{\operatorname{arg\,max}} \{f(x_1^e, x^n; w), f(x_2^e, x^n; w), \dots, f(x_A^e, x^n; w)\} \quad (1)$$

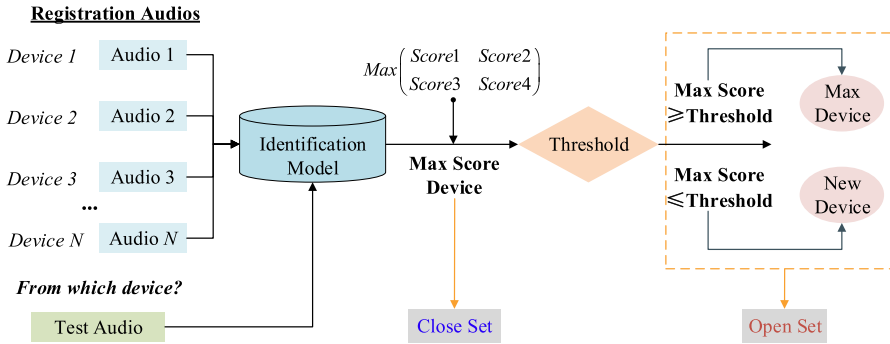


Fig. 1 Framework of the digital audio source identification

Here,  $x^n$  represents the input digital audio,  $w$  denotes the model's backend parameters,  $A$  is the number of registered devices ( $A > 1$ ), and  $a^*$  corresponds to the identified device. If the input digital audio always corresponds to one of the  $A$  devices registered in the database, the problem is considered a closed-set identification problem. Conversely, it is an open-set identification problem.

The overall framework illustrating the digital audio source identification problem is depicted in Fig. 1. By clarifying the problem statement and introducing relevant symbols, we have set the foundation for further discussions and analysis in subsequent sections of this paper.

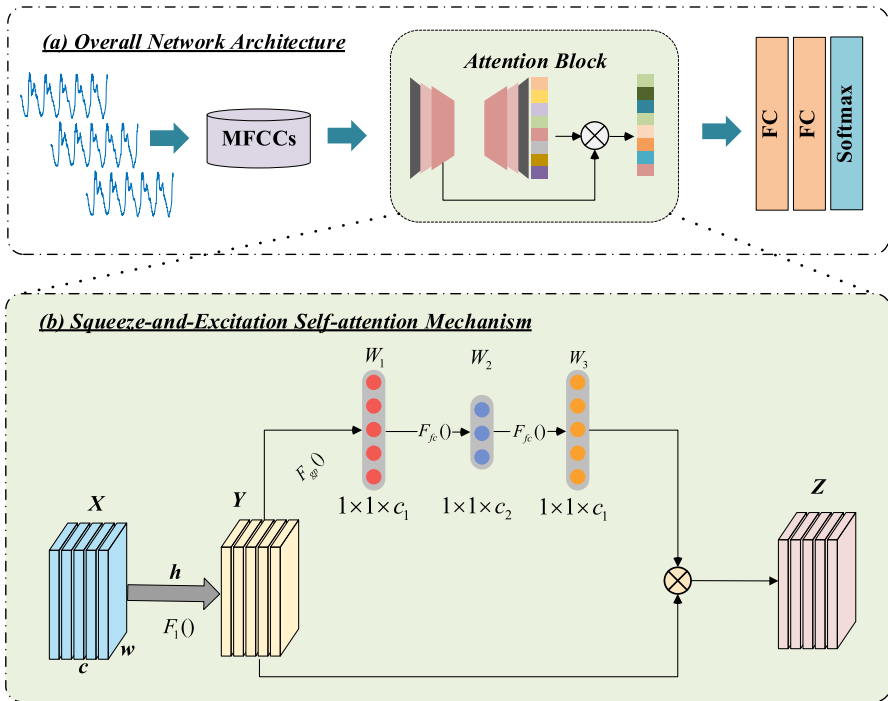
## 4 Methods

Previous research has demonstrated that deep neural networks possess excellent feature learning capabilities and strong representation modeling capabilities in the field of digital audio source recognition. Consequently, this study focuses on addressing the problem of representation modeling. However, practical applications of digital audio source recognition often encounter two primary challenges:

- **The Few-shot Problem:** It is impractical to exhaustively enumerate the sample sets of various digital audio devices, making it impossible to fully characterize the feature models of all devices. Consequently, when encountering a new audio source, the original representation model becomes inapplicable, requiring significant effort to retrain the model.
- **The Few-shot Sets Problem:** While deep neural networks exhibit good feature extraction and characterization abilities, their performance heavily relies on a large number of training data sets. When the sample size of the training data set is small, the model's performance is significantly impacted.

To address these challenges, this section employs transfer learning [27], a theoretical approach, to develop a network model based on deep neural networks suitable for digital audio source recognition.

Considering the significant differences in sparsity among features extracted from audio signals generated by different types of devices and the lack of additional char-



**Fig. 2** **a** Overall network architecture based on Squeeze-and-Excitation self-attention mechanism, in which the number of residual or convolutional blocks can be adjusted; **b** Schematic diagram of Squeeze-and-Excitation self-attention mechanism blocks

acterization information available in practical applications, this study enhances the robustness of the representation model by incorporating a self-attention mechanism into the construction of deep neural networks. Specifically, this chapter investigates the performance of the Squeeze-and-Excitation self-attention mechanism [17] for MFCC feature representation modeling and the impact of transfer learning based on deep residual networks. The following four aspects are described in this section:

- **Squeeze-and-Excitation Self-Attention Mechanism Module:** This module explores the effectiveness of the Squeeze-and-Excitation self-attention mechanism for enhancing MFCC feature representation modeling.
- **Residual Module:** This module focuses on the application of deep residual networks and their role in improving transfer learning performance.
- **Transfer Learning Module:** This module delves into the utilization of transfer learning techniques to enhance the digital audio source recognition model.
- **Decision Module:** This module outlines the decision-making process within the overall network model.

By addressing these aspects, this section provides a comprehensive overview of the methods employed in this study, setting the stage for the subsequent analysis and results.



#### 4.1 Squeeze-and-Excitation Based Self-Attention Mechanism Block

As illustrated in Fig. 2a, this study uses a deep neural network with a SE self-attention mechanism to process MFCC features, yielding more representative deep embedding features. These features are subsequently reduced in dimensionality through two fully connected layers, with classification decisions made via a Softmax classifier. This paper chooses to perform deep feature extraction on MFCC features rather than directly processing raw audio data, as MFCCs, being frequency-based features, effectively capture the frequency components of speech. By mapping the spectrum onto the Mel scale, MFCCs simulate the auditory characteristics of the human ear, showing greater sensitivity to low frequencies and lower resolution for high frequencies. Subsequently, logarithmic operations compress the spectrum, reducing the dynamic range of amplitude and enhancing noise resistance. Finally, the Mel spectrum is converted into cepstral coefficients via discrete cosine transform, removing spectral correlations, which makes the features more compact and facilitates more efficient modeling.

Figure 2b presents a schematic diagram of the basic SE-block in the residual network. The SE-block self-attention mechanism assigns varying weights to each two-dimensional feature map, amplifying critical features while diminishing less significant ones. This mechanism enhances the model's sensitivity to channel features, thereby improving the specificity and accuracy of feature extraction.

The implementation of the SE block involves two main parts. Firstly, the convolutional feature extraction transformation ( $X \rightarrow Y$ ) is carried out. Assuming that the input feature layer  $X$  consists of  $c$  channels, represented as  $X = [x_1, x_2, \dots, x_c]$ , where  $x_n \in R^{w \times h}$ , the convolution kernel  $FL = [fl_1, fl_2, \dots, fl_c]$  is obtained through  $m$  convolution filtering. This process generates the output feature layer  $Y = [y_1, y_2, \dots, y_c]$ , where  $y_n \in R^{w \times h}$ , and is computed as follows:

$$y_n = F_1(X) = \partial \left( \sum_{i=1}^c fl_n * x_i \right), \quad n \in [1, c] \quad (2)$$

Here,  $*$  denotes the convolution operation and  $\partial$  denotes the activation function.  $fl_n$  represents the two-dimensional convolution kernel acting on the corresponding channel of  $X$ . This convolutional operation extracts edge features from each layer while also fusing information across feature layers to capture spatial correlation.

The second part of the SE block involves the attention mechanism transformation ( $Y \rightarrow Z$ ). To obtain global information rather than local information for different feature layers, the global information of the feature layer  $Y$  is compressed into a vector using global averaging pooling. This operation yields a  $1 \times 1 \times c_1$ -dimensional weight initialization vector  $W_1 = [w_{1,1}, w_{1,2}, \dots, w_{1,c_1}]$ . The values of  $w_{1,c_1}$  are computed as follows:

$$w_{1,n} = F_{gp}(y_n) = \frac{1}{h \times w} \sum_{i=1}^h \sum_{j=1}^w y_n(i, j), \quad n \in [1, c_1] \quad (3)$$

Next, the weight initialization vector is transformed through two fully connected layers, resulting in the final vector of feature layer channel weights, denoted as  $W_3 = [w_{3,1}, w_{3,2}, \dots, w_{3,c_1}]$ , where

$$W_3 = F_{fc}(F_{fc}(W_1)) = \partial(a_2 \times \partial(a_1 \times W_1 + b_1) + b_2) \quad (4)$$

The weights and biases of the fully connected layers are denoted as  $a_1, b_1, a_2$ , and  $b_2$ , respectively. Specifically,  $a_1$  is a weight matrix of size  $R^{c_2 \times c_1}$ ,  $b_1$  is a bias matrix of size  $R^{1 \times c_2}$ ,  $a_2$  is a weight matrix of size  $R^{c_1 \times c_2}$ , and  $b_2$  is a bias matrix of size  $R^{1 \times c_1}$ . In this experiment, to introduce nonlinearity in the weight transformation process, the activation function  $\partial$  is chosen as the sigmoid function. It should be noted that  $c_2 < c_1$ .

Afterward, the output  $Z$  of the SE block is obtained by element-wise multiplication of each channel of the feature mapping  $y_n$  with the corresponding weight  $w_{3,n}$ , given by:

$$z_n = w_{3,n} \cdot y_n \quad (5)$$

Here,  $Z = \{z_1, z_2, \dots, z_c\}$ , and  $z_n \in R^{w \times h}$ . The resulting  $Z$  represents the output of the SE block. To provide a clearer overview of the overall workflow of the experimental network, we summarize the algorithm in Algorithm 1.

---

### Algorithm 1 :SE-block self-attention

---

**Require:** Input Data:  $X$

**Ensure:** Layer Outputs:  $Z$

- 1: Convolution:
    - 2:  $Y = F_1(x)$ ;
  - 3: Global Mean Pooling:
    - 4:  $W_1 = F_{gp}(Y)$ ;
  - 5: Fully Connected Neural Network:
    - 6:  $W_2 = F_{fc}(W_1)$ ;
    - 7:  $W_3 = F_{fc}(W_2)$ ;
  - 7: Multiplier:
    - 8:  $Z = W_3 \times Y$ ;
  - 9: Repeat the previous steps to stack the SE-blocks with an output of  $h$ ;
  - 10: Through two fully connected layers:
    - 11:  $h_1 = F_{fc}(h)$ ;
    - 12:  $h_2 = F_{fc}(h_1)$ ;
  - 12: The final decision is reached through the Softmax layer:
    - 13:  $prediction = Softmax(h_2)$ ;
-

## 4.2 Residual Structure Block

In the realm of speech processing, the development of traditional convolutional neural networks has demonstrated that increasing the network depth generally enhances the model's representational power and feature extraction capabilities. However, a phenomenon known as degradation arises when the network becomes too deep. This degradation manifests as weakened representational ability and feature extraction as depth increases. The two main factors contributing to this issue are gradient explosion/vanishing and the loss of informative yet weak characteristics during feature extraction. While techniques such as normalization layers can alleviate gradient-related problems, addressing the information loss remains challenging.

To overcome the information loss problem, researchers discovered that mapping the additional layers of a network with a constant mapping prevents the occurrence of information loss. Motivated by this insight, Kaiming He and his team introduced the Residual Network (ResNet) in 2015 [15]. ResNet comprises stacked residual modules with shortcut connections between them. By incorporating shortcut connections, ResNet can be viewed as a composition of relatively shallow networks. Veit et al. [34] further demonstrated that individual residual blocks within ResNet can be removed without significantly affecting the overall performance, highlighting that deep residual networks are essentially combinations of shallower residual modules.

The core elements of ResNet are the residual modules and shortcut connections. Figure 3 illustrates the schematic diagram of a residual module in ResNet. The module is defined as follows:

$$H(x) = F(x, \{W_i\}) + x + b \quad (6)$$

Here,  $x$  and  $H(x)$  denote the input and output vectors, respectively, and  $b$  represents the bias. The function  $F(x, \{W_i\})$  denotes the learned residual mapping. The right side of Fig. 3 depicts the shortcut connection in ResNet, where the constant mapping  $x \rightarrow x$  is used. Instead of directly learning the target mapping, the residual neural network learns the residual  $F(x) = H(x) - x$ . This residual mapping consists of two components: a nonlinear mapping  $F(x)$  and a linear direct mapping  $x \rightarrow x$ . When the nonlinear mapping is optimal, the network learning process automatically sets the weight of the nonlinear mapping to 0 and vice versa. This connection structure allows the network to build very deep networks by combining the two mapping approaches as constant transformations when the nonlinear mapping is optimal. Additionally, it enables the network to selectively learn features during the weight learning process, discarding redundant features.

There are two design variations for the ResNet module, as shown in Fig. 4a and b. The structure depicted in Fig. 4a is typically used for shallower networks like ResNet34, whereas the structure in Fig. 4b is employed for deeper networks such as ResNet50/101/152. The structure in Fig. 4b replaces two convolutional layers with a combination of two  $1 \times 1$  convolutional layers and one  $3 \times 3$  convolutional layer. This modification reduces the number of parameters and makes it more suitable for deeper networks compared to the structure in Fig. 4a. The structure in Fig. 4b first reduces dimensionality using a  $1 \times 1$  convolutional layer to reduce computation and

Fig. 3 Schematic diagram of the residual module of ResNet

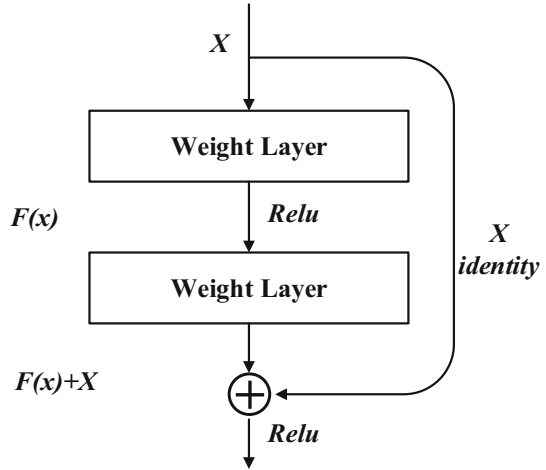
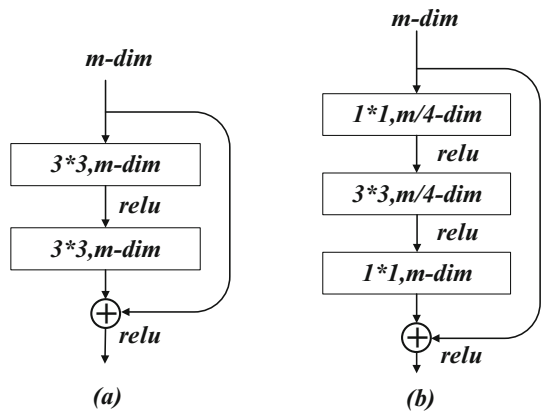


Fig. 4 Structure of the minimum module of the residual network



then further reduces it using another  $1 \times 1$  convolutional layer to maintain accuracy while reducing computation.

The learning iteration process in ResNet can be described as follows:  $F = W_2(\sigma(W_1x))$ , where  $\sigma$  represents the activation function, often employing the ReLU function.

The recurrence relation equation of the residual network is given by:

$$x_{l+1} = x_l + F(x_l, W_l) \tag{7}$$

$$x_{l+2} = x_{l+1} + F(x_{l+1}, W_{l+1}) = x_l + F(x_l, W_l) + F(x_{l+1}, W_{l+1}) \tag{8}$$

Equation (9) represents the general form of the recurrence relation in the residual network:

$$x_l = x_l + \sum_{i=l}^{L-1} F(x_i, W_i) \tag{9}$$

When a dimensional change occurs, the linear change in Eq. (10) can address the input–output mismatch:

$$x_{l+1} = Wx_l + F(x_l, W_l) \quad (10)$$

In ResNet, a batch normalization layer is added to the network between the output of the convolutional layer and the activation layer. Increasing the network depth leads to changes in the distribution of input values within the hidden layers. This shift can cause the data to move outside the sensitive interval of the activation function, resulting in slow gradient changes or gradient disappearance during error propagation. To counteract this, batch normalization is employed to bring the data back to the sensitive interval of the activation function, ensuring faster convergence and gradient preservation. However, applying batch normalization solely within the linear region of the activation function is not ideal for deep networks. To address this, a scale factor and offset parameters are introduced in batch normalization. These parameters not only bring the data back to the linear sensitive region but also shift the overall data slightly to the left and right, preventing it from adhering strictly to the standard variance. This adjustment satisfies the requirement for nonlinear variation.

### 4.3 Transfer Learning Block

In traditional classification learning tasks, a large number of labeled training samples are typically required to train a model for accurately classifying test data. However, in real-world scenarios, the availability of labeled samples is often limited or compromised due to environmental noise and other interferences. Consequently, the study of transfer learning becomes crucial for addressing few-shot learning challenges.

In few-shot learning tasks, a common approach is to employ transfer learning through model fine-tuning to train classification models. The process begins with pre-training a model on large-scale data, where a model is constructed in the source domain  $S_A$  using the data in the source domain  $D_{S_A} = \{(x_{S_1}, y_{S_1}), \dots, (x_{S_n}, y_{S_n})\}$ , with  $x_{S_i}$  representing the data sample in the source domain and  $y_{S_i}$  denoting the corresponding label.

$$F_{S_A}(\cdot) = \arg \min L(\cdot) \quad (11)$$

Following the pre-training phase, the parameters of the fully connected layer or the top layers of the network are fine-tuned using the target small sample data. In other words, the prediction model is obtained by training the target domain data  $D_{T_A} = \{(x_{T_1}, y_{T_1}), \dots, (x_{T_n}, y_{T_n})\}$  with fewer samples in the target domain  $T_A$ , where  $x_{T_i}$  represents the data sample in the target domain and  $y_{T_i}$  denotes the corresponding label. Typically, the learning rate used in the pre-training process is larger than the learning rate used in the target network. This allows the network to fine-tune at a slower pace, with the parameters of the pre-training model serving as initial values for the target network. As a result, the target network is no longer trained from a random seed starting point. The specific operations and comparisons of the network layers will

be discussed in detail in Sects. 5.5.4 and 5.5.5 within the context of the experimental design.

#### 4.4 Decision Block

In traditional regression models, the MSE error function is commonly used to calculate the model loss and serve as a training objective. The MSE is defined as:

$$loss = (\hat{y} - y)^2 \quad (12)$$

Here,  $\hat{y}$  represents the true label of the data, and  $y$  is the output of the data set through the model. Geometrically, this loss measures the spatial distance between the true labels and the model output, with the goal of minimizing this distance to achieve accurate regression.

For classification problems, optimizing the true labels and model outputs through minimizing geometric spatial distance is not feasible. This is because in classification problems, both the true labels and the model outputs are represented as probability distributions. Consequently, a loss function that captures the variability of the distribution is needed. Cross-entropy loss is a computational method commonly used to measure distribution variability in binary classification problems, and it is defined as:

$$loss = -[y \log \hat{y} + (1 - y) \log(1 - \hat{y})] \quad (13)$$

In the case of multi-classification problems, cross-entropy loss can still be used to measure distribution variability. However, it is important to note that in multi-classification problems, the number of classification labels is no longer limited to two, and there are multiple output distributions. Therefore, the specification of the results requires multi-classification before optimization. The desired specifications are: (1) all output distributions should have positive results, and (2) the sum of all output probability distributions should equal 1. The conventional Sigmoid layer does not satisfy these requirements. Hence, a Softmax layer is used before the output layer, and it is calculated as follows:

$$P(y = i) = \frac{e^{z_i}}{\sum_{j=0}^{K-1} e^{z_j}}, \quad i \in \{0, \dots, K - 1\} \quad (14)$$

Here,  $z^i \in \mathbb{R}^K$  represents the output of the last linear layer. The formula employs the natural constant as the base of the exponential operation, ensuring positive results and satisfying condition (1). The denominator represents the summation, ensuring that the sum of all results equals 1 and satisfies condition (2).

When calculating the cross-entropy loss in multi-classification problems, since the labels have multiple 0 terms, the loss equation becomes:

$$loss(\hat{y}, y) = -y \log \hat{y} \quad (15)$$

Minimizing this loss allows for decision-making based on the input model's data.

**Table 2** Descriptions of controlled-conditions and uncontrolled-conditions datasets

Datasets	Time	Equipment Source Type & Size	Recording environments	Recording duration (minutes)
Controlled-Conditions	2018	31 Mobile Phones	4 different locations	318 min
Uncontrolled-Conditions	2018	141 Mobile Phones	Same quiet environment	141,100 min

## 5 Experimental Results and Analysis

### 5.1 Datasets

This study utilized two datasets, Uncontrolled-Conditions Dataset [25] and CCNU Mobile Dataset [45], in experiments comparing with baseline systems, exploring the SE attention mechanism, and investigating transfer learning. Luo et al. [25] recorded both the Controlled-Conditions Dataset and the Uncontrolled-Conditions Dataset. The Controlled-Conditions dataset comprised recordings from 16 different environments across four settings (two offices, a hall, and a station) and four speakers. Each recording was 8 min in duration, with a sample rate of 44.1 kHz and a quantization rate of 16 bits. The recordings were conducted using the same speaker playback and involved 31 cell phones in five separate batches. However, as the research in the field of digital audio source identification progressed, the original small-scale dataset no longer sufficed for the study requirements. Consequently, Luo et al. recorded an Uncontrolled-Conditions Dataset that involved 141 devices. Each device was recorded in the same quiet environment, resulting in recordings of 10 min in duration, a sample rate of 44.1 kHz, and a quantization rate of 16 bits for the speech data. The relevant datasets are summarized in Table 2.

The speech data from the original CCNU Mobile Dataset<sup>1</sup> was recorded using 45 different devices, including eight different brands (including a small number of tablets). The CCNU Mobile Dataset specifically consists of pure TIMIT data that was not transcribed from other devices. Due to the limited sample data in the TIMIT dataset, recording the entire dataset as a whole was not feasible. Instead, all the training data in the TIMIT dataset were selected and merged in order to form a single long corpus of approximately 110 min in length.

Subsequently, using the active speech detection method, the silent segments before and after the recorded long speech data were removed. Finally, to facilitate subsequent studies, the long speech data were split into 642 small sample segments, each approximately 10 s in duration. The splitting was done based on the number and duration of the TIMIT corpus, following the order of the samples at the time of merging. The brand and model information of the recording devices in the CCNU Mobile Dataset are presented in Table 3.

<sup>1</sup> [https://github.com/CCNUZFW/CCNU\\_Mobile\\_dataset](https://github.com/CCNUZFW/CCNU_Mobile_dataset).

**Table 3** Brand and model of recording devices in CCNU mobile dataset

Brands	Models
APPLE	iPhone6,iPhone6s,iPhone SE,iPad7,iPhone7p,iPhoneX,air2,air1
HUAWEI	tag-a100,nova,novo2s,nova3e,honor7x,honor8,honorV8,honor9,honor10,p10,p20
XIAOMI	mi2s,note3,mi5,mi8,mi8se,mix2,redmi-Note4x,redmi-3 S
VIVO	y11t,x3f,x7
ZTE	c880a,g719c
SAMSUNG	Sph-d710,s8
OPPO	r9s
NUBIA	z11

## 5.2 Evaluation Metrics

### 5.2.1 Confusion Matrix

In classification tasks, prediction outcomes can be categorized into four distinct types: a) True Positive (TP); b) False Positive (FP); c) True Negative (TN); and d) False Negative (FN).

For a multi-class classification model, considering the first class as an example (denoted by subscript 1), TP refers to the count of instances that genuinely belong to the first class and are accurately classified as such. FP corresponds to the number of instances that do not belong to the first class but are incorrectly classified as members of it. TN represents the count of instances that do not belong to the first class and are correctly excluded from it. FN indicates the number of instances that truly belong to the first class but are erroneously classified as not belonging to it.

### 5.2.2 Four Metrics

This paper adopts relevant evaluation metrics based on the confusion matrix to measure model performance, with class 1 as the focus of evaluation. The following metrics are employed:

(1) **Accuracy Rate** This metric represents the ratio of correctly predicted results to the total number of predictions. It is defined as follows:

$$Acc = \frac{\text{All accurate predictions}}{\text{Number of predictions}} \quad (16)$$

(2) **Recall Rate** The recall rate calculates the percentage of correct predictions for a specific class among all samples of that class. This indicator assesses the completeness of the predictions, regardless of false negatives. It is defined as follows:

$$Recall(class1) = \frac{TP_1}{TP_1 + FN_1} \quad (17)$$



(3) **Precision Rate** The precision rate calculates the percentage of correct predictions for a specific class among all samples predicted to belong to that class. This metric focuses on the accuracy of the predictions, regardless of false positives. It is defined as follows:

$$Precision(class1) = \frac{TP_1}{TP_1 + FP_1} \quad (18)$$

(4) **F-Score** The F-score is a weighted combination of precision and recall rates. It provides an overall measure of model performance, with a higher value indicating better performance. The F-Score is defined as follows:

$$F_\beta(class1) = (1 + \beta^2) \times \frac{Precision(class1) \times Recall(class1)}{\beta^2 \times Precision(class1) + Recall(class1)} \quad (19)$$

Here,  $\beta$  represents the weight given to recall and precision, reflecting their relative importance. In the context of the digital audio source identification problem, which involves multi-classification, the maximum probability of correct labeling is achieved through Softmax with a fully connected layer. When evaluating the overall model, the scores for all classes are calculated and then averaged. For the F-Score, the F1-score with  $\beta = 1$  is selected for this experiment.

### 5.3 Baselines

To assess the overall performance of the system proposed in this paper, we conducted a comparative experiment against four classic methods: i-vector+SVM [30], Band Energy Difference(BED)+SVM [25], MFCC+SVM [13], and GSV+CNN [4]. The characteristics of these four baseline systems are summarized as follows:

(1) **i-vector + SVM [30]** The i-vector reduces the dimensionality by obtaining the speech feature vector of the high-dimensional target device source, projecting it in the subspace, and using factor analysis to eliminate the factors that put redundancy to obtain the low-dimensional feature vector.

(2) **BED + SVM [25]** The method, based on spectral feature extraction, calculates the baseband energy difference to intuitively describe differences in device sources while effectively reducing computational overhead. During the BED feature extraction process, 256 Fourier spectrum sampling points are used. The differential operation results in a baseband energy difference feature with dimensions of (1, 127).

(3) **MFCC + SVM [13]** This method utilizes MFCC, widely used in audio recognition, as the input audio features and employs SVM as the classification model.

(4) **GSV + CNN [4]** This method combines the representative GSV features with a CNN, exemplifying a deep learning approach. GSV features are represented as a two-dimensional matrix and are processed using a CNN with a convolutional kernel size of 33. Batch normalization layers are utilized between convolutional layers to process the feature maps effectively.

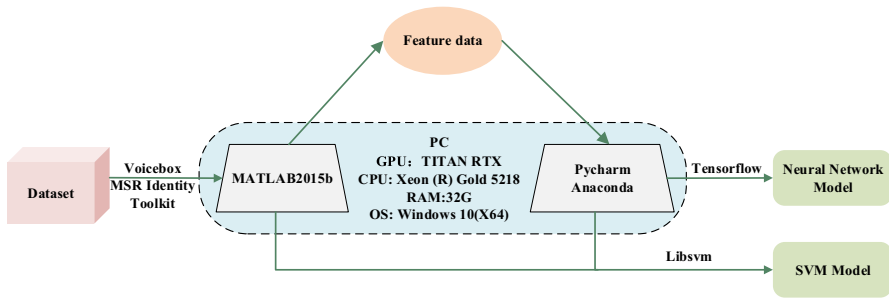


Fig. 5 Experimental platform setup

## 5.4 Experimental Platform Setup

To facilitate the experiments conducted in this study, we established the experimental platform shown in Fig. 5. The experiments were performed on a desktop computer running the Win10 operating system. The computer was equipped with a TITAN RTX GPU, a Xeon (R) Gold 5218 CPU, and 32 GB of RAM. The software used in the experiments included MATLAB 2015b, PyCharm, and Anaconda. Additionally, various toolkits such as Voicebox, MSR Identity Toolkit, Libsvm, and TensorFlow were utilized.

## 5.5 Experimental Results

In this section, we set two main objectives while investigating digital audio source identification methods using deep residual network transfer learning. The first objective was to explore the effectiveness of the SE self-attention mechanism in the digital audio domain. The second objective was to determine whether transfer learning could enhance training performance with small sample sizes in this field. To achieve these goals, we designed experiments comparing our approach with a baseline system and conducted a detailed study on the SE-Block self-attention mechanism to assess its practical impact. This study also examined the interplay between the introduced self-attention mechanism and the chosen network architecture. Additionally, we designed experiments to validate the effectiveness of the transfer model, examining both global and partial transfer scenarios. These experiments demonstrated the applicability of transfer learning in digital audio source identification and explored its effects on training with small sample datasets.

### 5.5.1 Comparative Experiments with Four Baseline Methods

The objective of this experiment is to compare the performance of the proposed method with that of the baseline systems. To facilitate a clear comparison of the various performance metrics, this set of experiments utilized the CCNU Mobile Dataset for a 45-class classification task. The dataset was divided into training and validation sets in a 3:2 ratio. The experimental results are presented in Table 4.

**Table 4** Experimental results of performance comparison with baseline methods

Method	Accuracy	Recall	Precision	F1-Score
i-vector+SVM [30]	64.6%±1.2%	64.6%±1.2%	65.9%±1.2%	64.7%±1.2%
MFCC+SVM [13]	86.8%±0.2%	86.8%±0.2%	86.9%±0.2%	86.9%±0.2%
BED+SVM [25]	93.4%±0.5%	93.4%±0.5%	93.7%±0.5%	93.3%±0.5%
GSV+CNN [4]	92.9%±0.5%	92.9%±0.5%	93.6%±0.5%	92.8%±0.5%
Ours	<b>95.2%±0.4%</b>	<b>95.2%±0.4%</b>	<b>95.4%±0.4%</b>	<b>95.1%±0.4%</b>

Best performance for digital audio source recognition are given in bold

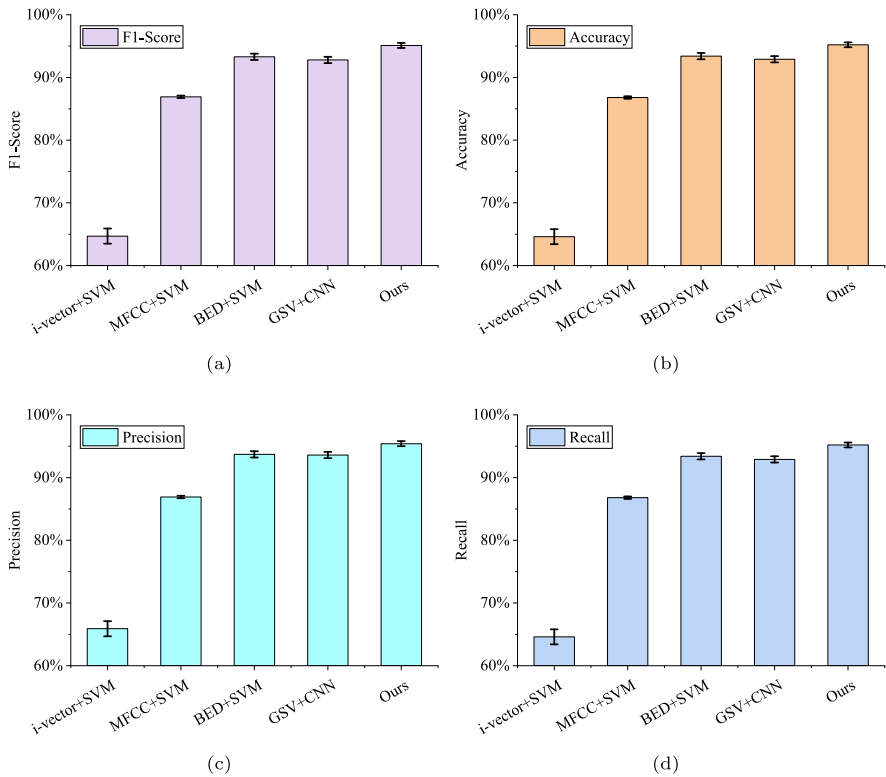
As presented in Fig. 6, our method achieved an accuracy rate of 95.2%, representing an improvement of 1.8% to 30.6% over the baseline methods. The i-vector + SVM method exhibited lower performance metrics, indicating limitations in the SVM's ability to map complex features. In contrast, the MFCC + SVM method showed improved accuracy and F1-Score, reaching 86.8% and 86.9%, respectively, demonstrating the effectiveness of MFCC in extracting speech features. However, the BED + SVM method further enhanced performance, likely due to the stability and accuracy of BED features in distinguishing speaker characteristics. Nonetheless, this method has higher computational complexity and sensitivity to specific acoustic environments and equipment, which may limit the generalization of features. In the specific application scenario under study, MFCC provided more robust and consistent performance.

For neural network models, the GSV + CNN method demonstrated strong performance, with an accuracy of 92.9% and an F1-Score of 92.8%, highlighting the advantages of CNN in handling complex features, as in Fig. 6. However, this method did not achieve the highest performance. In contrast, our proposed method excelled across all evaluation metrics, indicating that our approach enables more representative deep embedding of MFCC features, demonstrating the method's feasibility. Compared to traditional methods, our model more effectively captures and distinguishes complex speech features, underscoring the critical role of selecting appropriate feature extraction techniques and deep learning architectures in enhancing the accuracy and reliability of speech recognition.

### 5.5.2 Hyperparameter Sensitivity Analysis

In this experiment, we deliberately deviated from optimal values for three hyperparameters, which are learning rate, dropout rate, and batch size, to evaluate how sensitive the model is to these changes. Table 5 provides a detailed account of the resulting changes in model performance under each adjustment. This table demonstrates that hyperparameter tuning was performed to ensure the model's performance was both optimized and stable. This process highlights how adjusting different hyperparameters influences the model's final performance.

As shown in Fig. 7, the Dropout Rate hyperparameter was varied dynamically between 0.5 and 0.9, in increments of 0.1, while keeping Batch Size and Learning Rate constant. According to the results, the optimal Dropout Rate was found to be 0.8, which we have adopted as the best setting for our method.



**Fig. 6** Comparison with 4 baseline methods. **a** is the  $F1$  of the baseline methods and the proposed method. **b** is the  $ACC$  of the baseline methods and the proposed method. **c** is the  $Precision$  of the baseline methods and the proposed method. **d** is the  $Recall$  of the baseline methods and the proposed method

Figure 8 presents a scenario where the Dropout Rate and Learning Rate were fixed while the Batch Size was varied dynamically between 40 and 120, in increments of 20. The experimental data indicate that a Batch Size of 60 yielded the best results, which we have selected as the optimal setting.

In Fig. 9, by fixing the Dropout Rate and Batch Size and dynamically adjusting the Learning Rate from  $1e-5$  to  $3e-4$ , we identified that a Learning Rate of  $1e-4$  provided the best performance. Consequently, we have set this rate as the optimal choice for our method.

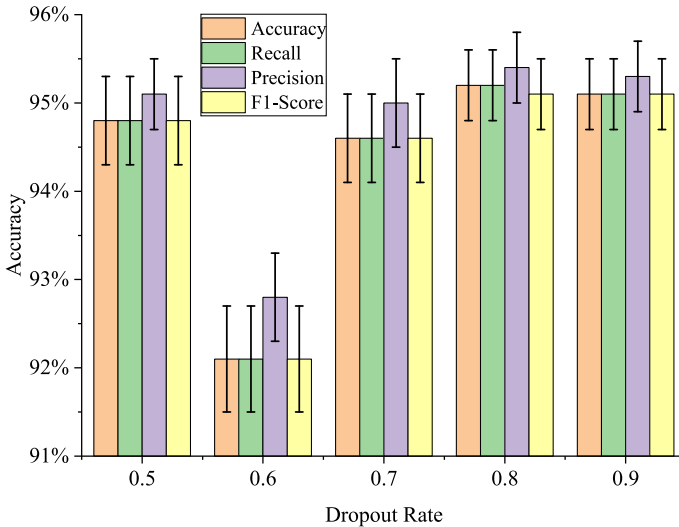
### 5.5.3 Validation of the Effectiveness of SE Self-Attention Mechanism

To provide a more detailed analysis of the model's performance on a smaller category set while managing experimental complexity, this set of experiments utilized the Uncontrolled-Conditions Dataset for a 20-class classification task. This design effectively minimizes the impact of inter-class similarity on the model, enabling a clearer evaluation of its nuanced capabilities in distinguishing among relatively fewer categories. Additionally, this approach allows us to assess the differences in

**Table 5** Experimental results of hyperparametric sensitivity analysis

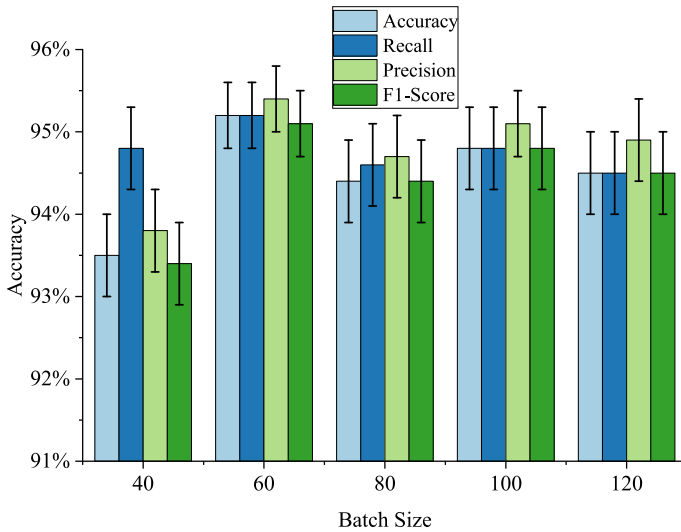
Hyperparameters	Adjustment	Evaluation Metrics			
		Accuracy	Recall	Precision	F1-Score
Dropout Rate	0.5	94.8%±0.5%	94.8%±0.5%	95.1%±0.4%	94.8%±0.5%
	0.6	92.1%±0.6%	92.1%±0.6%	92.8%±0.5%	92.1%±0.6%
	0.7	94.6%±0.5%	94.6%±0.5%	95.0%±0.5%	94.6%±0.5%
	<b>0.8</b>	<b>95.2%±0.4%</b>	<b>95.2%±0.4%</b>	<b>95.4%±0.4%</b>	<b>95.1%±0.4%</b>
	0.9	95.1%±0.4%	95.1%±0.4%	95.3%±0.4%	95.1%±0.4%
Batch Size	40	93.5%±0.5%	93.5%±0.5%	93.8%±0.5%	93.4%±0.5%
	<b>60</b>	<b>95.2%±0.4%</b>	<b>95.2%±0.4%</b>	<b>95.4%±0.4%</b>	<b>95.1%±0.4%</b>
	80	94.4%±0.5%	94.4%±0.5%	94.7%±0.5%	94.4%±0.5%
	100	94.8%±0.5%	94.8%±0.5%	95.1%±0.4%	94.8%±0.5%
	120	94.5%±0.5%	94.5%±0.5%	94.9%±0.5%	94.5%±0.5%
	Learning Rate	1e-5	93.9%±0.5%	93.9%±0.5%	94.0%±0.5%
	<b>1e-4</b>	<b>95.2%±0.4%</b>	<b>95.2%±0.4%</b>	<b>95.4%±0.4%</b>	<b>95.1%±0.4%</b>
	1.5e-4	94.9%±0.5%	94.9%±0.5%	95.2%±0.4%	94.9%±0.5%
	2e-4	93.1%±0.5%	93.1%±0.5%	93.8%±0.5%	93.2%±0.5%
	3e-4	93.0%±0.5%	93.0%±0.5%	94.0%±0.5%	93.0%±0.5%

Best performance for digital audio source recognition are given in bold

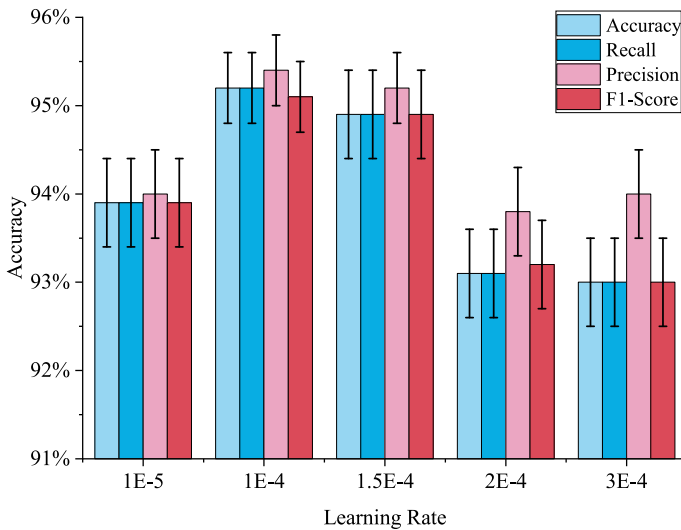


**Fig. 7** The sensitivity analysis experiments based on dropout rate

model performance between more challenging tasks (45-class classification) and simpler tasks (20-class classification), thus offering a comprehensive evaluation of the model’s robustness and generalization capabilities. In this chapter, we divided the dataset equally into a training set and a validation set, using a 1:1 ratio, to facilitate supervised training.



**Fig. 8** The sensitivity analysis experiments based on batch size



**Fig. 9** The sensitivity analysis experiments based on learning rate

In this experiment, we introduced the SE-Block self-attention mechanism into our proposed digital audio source recognition method, which is based on deep residual network transfer learning. Experiment Group I employed the original deep residual network, while Experiment Group II utilized the deep residual network with the SE-Block self-attention mechanism. To minimize the influence of irrelevant factors, we designed the network with the same nodes and layers. The detailed parameters are presented in Table 6. The audio short-time frame length was set to 16ms, with a frameshift of 50%. We applied the Hamming window, used 12 Mel filters, and obtained 39-dimensional

**Table 6** Network structure design parameters

Output size	Resnet	Res_SEblock
170 × 20	Conv 4964, stride 2	Conv 4964, stride 2
85 × 10	Maxpool 33, stride 2	Maxpool 33, stride 2
43 × 5	$\begin{bmatrix} 1 \times 1, & 64 \\ 3 \times 3, & 64 \\ 1 \times 1, & 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, & 64 \\ 1 \times 1, & 256 \end{bmatrix} \times 2 + \begin{bmatrix} 1 \times 1, & 64 \\ 3 \times 3, & 64 \\ 1 \times 1, & 256 \end{bmatrix} \times 1$
22 × 3	$\begin{bmatrix} 1 \times 1, & 128 \\ 3 \times 3 & 128 \\ 1 \times 1, & 512 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, & 128 \\ SEblock \\ 1 \times 1, & 512 \end{bmatrix} \times 5 + \begin{bmatrix} 1 \times 1, & 128 \\ 3 \times 3, & 128 \\ 1 \times 1, & 512 \end{bmatrix} \times 1$
11 × 2	$\begin{bmatrix} 1 \times 1, & 256 \\ 3 \times 3 & 256 \\ 1 \times 1, & 1024 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, & 256 \\ SEblock \\ 1 \times 1, & 1024 \end{bmatrix} \times 2 + \begin{bmatrix} 1 \times 1, & 256 \\ 3 \times 3, & 256 \\ 1 \times 1, & 1024 \end{bmatrix} \times 1$
	Global average pool <i>fc</i> 1000-d softmax	Global average pool <i>fc</i> 1000-d softmax
accuracy	90.7%	89.1%

inverse spectral coefficients, first-order difference coefficients, second-order difference coefficients, and logarithmic energy coefficients as the audio features. This paper chooses to perform deep feature extraction on MFCC features rather than directly processing raw audio data, as MFCCs, being frequency-based features, effectively capture the frequency components of speech. By mapping the spectrum onto the Mel scale, MFCCs simulate the auditory characteristics of the human ear, showing greater sensitivity to low frequencies and lower resolution for high frequencies. Subsequently, logarithmic operations compress the spectrum, reducing the dynamic range of amplitude and enhancing noise resistance. Finally, the Mel spectrum is converted into cepstral coefficients via discrete cosine transform, removing spectral correlations, which makes the features more compact and facilitates more efficient modeling.

The experimental results in Table 6 clearly show that the deep residual network with the SE self-attention mechanism did not significantly enhance recognition performance under the same number of network layers.

To further investigate this phenomenon, we conducted additional experiments to determine whether the observed performance decline was solely due to the SE-Block self-attention mechanism or its combination with the deep residual network. In this section, we performed ablation experiments by incorporating the SE-Block self-attention mechanism into a CNN. Table 7 outlines the experimental designs for two sets of network parameters. To minimize random factors, each experiment was repeated five times, and the average results for each metric were recorded as the final values in Table 8. Additionally, a 95% confidence interval was calculated using the normal approximation method. Furthermore, we generated a confusion matrix to provide a comprehensive comparison of the results.

Table 8 presents the experimental results of performance exploration based on the SE-Block self-attention module. We conducted five experiments on both the CNN and SE-CNN configurations and calculated the mean values of the five experimental runs. The evaluation metrics used include Accuracy, Recall, Precision, and F1-Score.

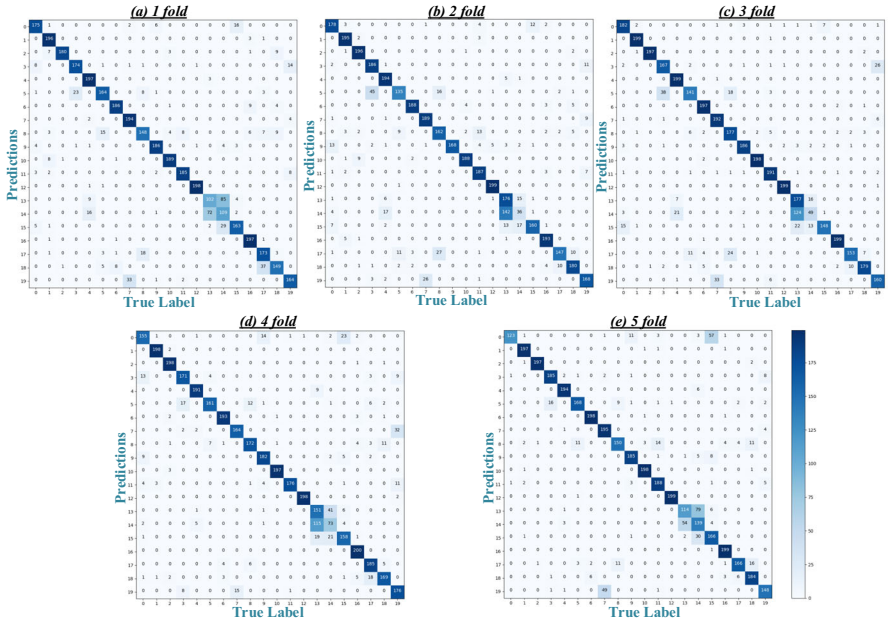
**Table 7** Design of convolutional network parameters

Output size	CNNNet	SE-CNNNet
$170 \times 20$	conv $5 \times 5, 6$ ; stride 1; SAME max pool $2 \times 2$ ; stride 2; SAME	
$85 \times 10$	conv $5 \times 5, 5, 16$ max pool $2 \times 2$	conv $5 \times 5, 16$ max pool $2 \times 2$
$43 \times 5$	conv $5 \times 5, 40$ max pool $2 \times 2$	conv $5 \times 5, 40$ max pool $2 \times 2$
$1 \times 1040$	Global average pool $215$ -d; $215$ -d; $f_c$ 1040-d	global average pool $16$ -d $16$ -d $16$ -d $f_c$ $6$ -d $6$ -d $f_c$ $16$ -d
$1 \times 600$	$1040$ -d $f_c$ $600$ -d	global average pool $40$ -d $40$ -d $f_c$ $16$ -d $16$ -d $f_c$ $40$ -d
$1 \times 20$	Softmax	



**Table 8** Experimental results of performance analysis based on SE-block self-attention mechanism

Network Configuration	Evaluation Metrics			
	Accuracy	Recall	Precision	F1-Score
CNNNet	86.5% ± 1.1%	86.5% ± 1.1%	87.0% ± 1.0%	86.3% ± 1.1%
SE-CNNNet	88.0% ± 1.0%	88.0% ± 1.0%	88.5% ± 1.0%	87.8% ± 1.0%

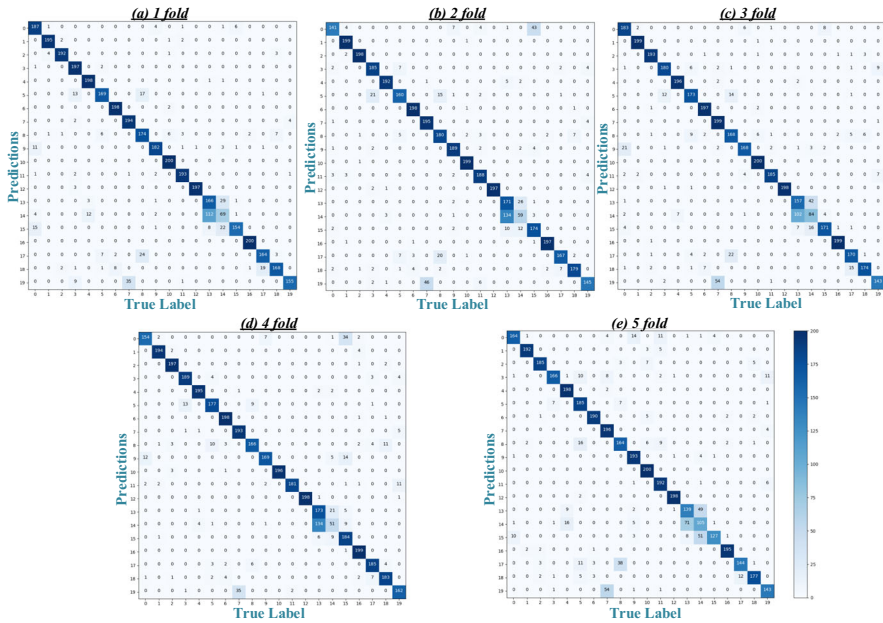


**Fig. 10** The confusion matrix of the results of the 5 experiments of the CNNNet experimental group

Furthermore, we plotted the confusion matrices to visually compare the results of the CNNNet and SE-CNNNet experimental groups. Figure 10 depicts the confusion matrix for the CNNNet experiments, while Fig. 11 illustrates the confusion matrix for the SE-CNNNet experiments.

The experimental results presented in Table 8 and the corresponding confusion matrices in Figs. 10 and 11 provide intuitive evidence regarding the impact of the SE-Block self-attention mechanism on the recognition performance of the CNN. We observe that the introduction of the SE-Block self-attention mechanism improves the recognition effect compared to the original CNN configuration. This finding suggests that the SE-Block self-attention mechanism enhances the representativeness of the convolutional neural network.

Based on the above experimental results, it is evident that the selection of an appropriate attention mechanism is crucial for different network structures and practical applications. Failing to choose the appropriate attention mechanism may lead to a decrease in the recognition performance of the network.



**Fig. 11** The confusion matrix of the 5 experimental results of the SE-CNNNet experimental group

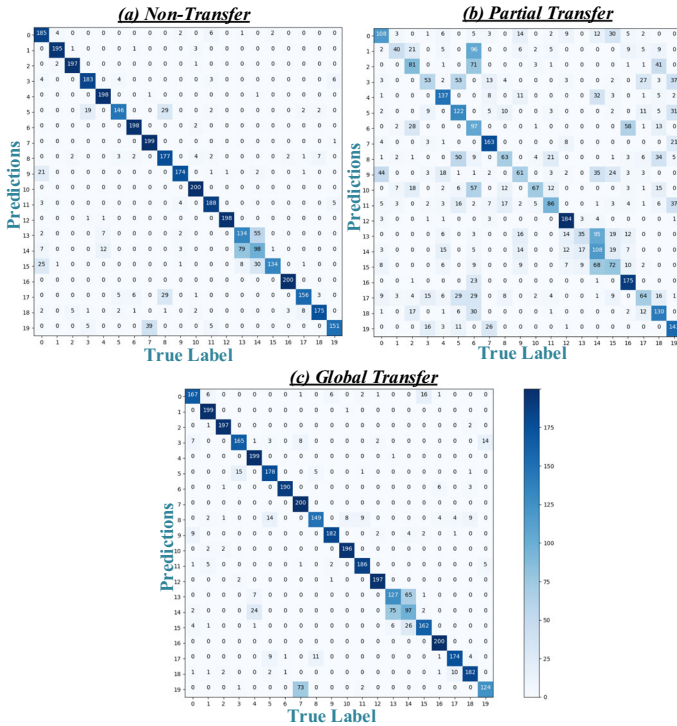
In summary, the experimental results, supported by the evaluation metrics and visualized through confusion matrices, provide strong evidence for the effectiveness of the SE-Block self-attention mechanism in enhancing the representativeness of the convolutional neural network. Despite each data sample being of short duration, all samples are sequentially fed into the model over multiple iterations during the training process. The model architecture utilizes a CNN to extract multi-level information through convolutional operations. As the network depth increases and the SE attention mechanism is introduced, the model can incrementally extract deep embeddings from the original data samples. These embeddings capture intricate relationships between the samples. Consequently, the proposed model can effectively manage information across various window lengths and scales, ensuring thorough extraction and analysis.

### 5.5.4 Validation of the Effectiveness of Transfer Learning

In the previous subsection, we demonstrated through experimental validation that incorporating the SE-Block self-attention mechanism improves the accuracy of digital audio source recognition by convolutional networks. Building upon this finding, we now investigate the effectiveness of transfer learning using a convolutional neural network based on the SE-Block self-attention mechanism. Transfer learning encompasses two main approaches in practical applications: feature-based representation learning and optimization based on initial network performance. In this subsection, we explore both transfer methods through experiments.

**Table 9** Comparison of experimental results for two transfer learning methods

Evaluation Metrics	Non-transferable	Partial	Global
Accuracy	87.1% ± 1.0%	49.7% ± 1.5%	86.8% ± 1.0%
Recall	87.1% ± 1.0%	49.7% ± 1.5%	86.8% ± 1.0%
Precision	87.7% ± 1.0%	53.5% ± 1.5%	87.1% ± 1.0%
F1-Score	87.0% ± 1.0%	48.5% ± 1.5%	86.6% ± 1.1%



**Fig. 12** Confusion matrix diagram for experimental results with three transfer levels

To begin the experimental investigation of transfer learning, we first train a base network model. The dataset selection process is described in Sect. 5.1. For consistency with the shape of the transferred network input, we extract and label the MFCC features from 20 devices in the CCNU Mobile Dataset for base network training. The experimental results obtained from the two transfer methods are presented in Table 9. The dataset division in this chapter follows the same approach as the experiments in the previous chapter, using a 1:1 ratio to separate the validation set for supervised training.

The experimental results in Table 9 and the confusion matrix for the three transfer levels shown in Fig. 12 reveal that using the base network as initial values yields significantly better results compared to fixing the parameters of the first few layers

for representation learning. This suggests that transfer learning has the capability of feature learning in the field of digital audio source recognition research. Moreover, it demonstrates that initializing the original model outperforms using the initial layers of the network for feature extraction, indicating that fixing the parameters in the early layers has a greater impact on recognition performance. However, the global optimization process of the network is more time-consuming.

From a theoretical perspective, transfer learning allows for a wider exploration of network parameter values in digital audio source recognition, thereby enhancing the network's generalization ability. Subsequently, adaptive learning optimization based on specific data can yield more accurate feature representation methods and precise recognition results. Surprisingly, our experimental results show that transfer learning slightly underperforms compared to non-transfer learning. This suggests that achieving higher recognition results through transfer learning requires a highly generalizable and pervasive base network with strong feature representation capabilities across different device classes in the field of digital audio source recognition. Additionally, the dataset used to train the base network should possess high generalizability and include various sources of variation.

### 5.5.5 Validation of the Effectiveness of Few-Shot Learning

In the previous subsection, we observed the positive effects of transfer learning on digital audio source recognition research through experiments comparing different transfer levels. In this section, we delve into the few-shot learning case, which is one of the practical problems that transfer learning can address.

To accurately represent the experimental data in the few-shot learning scenario, we made the following changes to the dataset: (1) We expanded the CCNU Mobile Dataset used for the base network from 20 devices to include all 45 devices. (2) The Uncontrolled-Conditions Dataset used for the transferable target network was expanded from 20 to 141 devices. (3) We reduced the Uncontrolled-Conditions Dataset used in the transferable target network from 400 to 40 speech clips per device and divided the training and test sets according to a 3:1 ratio. The validation set in the training data is still divided using a 1:1 ratio. This setup is suitable for cases where there are too few samples available for the specified data, but there is another dataset with a larger amount of information that can be used for transfer training.

Regarding the network architecture, we trained the base network with a large learning rate and saved it. The transfer target network was fine-tuned by reducing the learning rate and making the following architecture changes: (1) We removed the original softmax layer. (2) Two new fully connected layers and a softmax layer were added to accommodate the 141 devices case. (3) For the unchanged covariates, we used the values from the trained base network as initial values for training. The experimental results are presented in Table 10.

However, the purpose of this experiment is to investigate the utility of transfer learning in this case, and further exploration is required to enhance the performance. By examining the experimental data, we observed that the recognition accuracy improves when a base network with ample information is used as the baseline for transfer. Additionally, while the recall rate improves, the accuracy rate decreases. Combining this

**Table 10** Comparison of experimental results between the original network and transfer to target network using the base network

Evaluation Metrics	Original network	Add two new FC and Softmax
Accuracy	22.0% $\pm$ 1.3%	25.7% $\pm$ 1.4%
Recall	22.0% $\pm$ 1.3%	25.7% $\pm$ 1.4%
Precision	26.5% $\pm$ 1.4%	21.5% $\pm$ 1.3%
F1-Score	17.9% $\pm$ 1.2%	19.2% $\pm$ 1.2%
Training time (s)	186	278

**Table 11** Experimental results for further refinement of the target network

Evaluation metrics	Train the original two FCs and softmax
Accuracy	29.2% $\pm$ 1.4%
Recall	29.3% $\pm$ 1.4%
Precision	30.5% $\pm$ 1.4%
F1-Score	23.3% $\pm$ 1.3%
Training time (s)	189

observation with the evaluation metrics introduced in Sect. 5.2.2, we can infer that the transfer network performs better in terms of comprehensive coverage, despite the possibility of misclassification. In other words, when the number of devices increases (from 20 to 141) and the number of sample entries decreases (from 400 to 40), transfer and fine-tuning make the network more robust across all categories. Furthermore, the improved F1 score indicates better performance under the combined metric evaluation, which suggests that the decrease in accuracy, i.e., misclassification, is likely due to factors other than the transfer process. We will discuss this further in the next experimental group.

Analyzing the training time, we observed a considerable increase. Upon examining the reasons for this increase, we found that the original network's last three layers consisted of two Fully Connected (FC) networks and one softmax layer for classification. In the updated network, we added four FC layers for training based on the previous experiments, resulting in an excessive number of hyperparameters that affected the experimental results. To address this issue, we further refined the network by excluding the initial values for the last three layers of the original network (2 FC + 1 softmax), treating them as entirely new layers to be trained. The remaining layers still retained their initial values. The experimental results for the further refinement of the target network are presented in Table 11.

The experimental results demonstrate that the refined network further improves the model's performance. Additionally, the decrease in accuracy observed in the previous experimental group is eliminated, confirming that the accuracy decline resulted from the excessive number of hyperparameters rather than the transfer process itself.

In conclusion, the experiments conducted in the few-shot learning case indicate that transfer learning can be beneficial, particularly when using a base network with abundant information as the source for transfer. Although the accuracy may decrease

slightly, the overall performance in terms of comprehensive coverage and combined metric evaluation is enhanced. Moreover, refining the network by carefully managing the hyperparameters leads to further performance improvements without compromising accuracy. These findings shed light on the effective application of transfer learning in scenarios where limited data is available.

## 6 Conclusions

This study addressed key challenges in representation modeling of digital audio source recognition, specifically targeting incremental representation and the limitations of representation with few training samples. Our approach involved the application of transfer learning strategies within residual network frameworks and the incorporation of Squeeze-and-Excitation Blocks to bolster the robustness of the representation model. Our experimental findings indicate that the benefits of the self-attention mechanism vary depending on the specific network architecture employed. While self-attention can enhance model robustness, its effectiveness is not universally guaranteed and is highly dependent on the network's configuration. This highlights the nuanced impact of self-attention mechanisms within different architectural contexts. Moreover, our results confirm the viability of using transfer learning to train representation models effectively with limited data. By pre-training on similar, large-scale datasets, the model can be fine-tuned with smaller data sets, albeit requiring additional time for global network optimization. This approach shows promise in refining data with limited initial training samples. Despite the progress made, our study identifies several areas requiring further exploration. The experimental validations, while extensive, did not encompass a sufficient variety of datasets, particularly large batch datasets, which could further elucidate the dynamics of transfer learning in digital audio source recognition. Future research will aim to enhance the generalization capabilities of these models and improve recognition accuracy, particularly through fine-tuning in few-shot learning scenarios. In summary, our research advances the understanding of transfer learning's potential in digital audio source recognition and underscores the importance of tailored approaches in representation modeling for challenging scenarios. Continued exploration in this field is essential to fully leverage transfer learning techniques in audio processing, aiming to achieve broader applicability and more precise outcomes.

**Acknowledgements** The research work of this paper were supported by the National Natural Science Foundation of China (No. 62177022, 61901165, 61501199), Self-determined Research Funds of CCNU from the Colleges' Basic Research and Operation of MOE (No. CCNU24JC033), Natural Science Foundation of Hubei Province (No. 2022CFA007), and Wuhan Knowledge Innovation Project (No. 2022020801010258).

**Data Availability** Data will be made available on reasonable request.

## Declarations

**Conflict of interest** No potential conflict of interest was reported by the authors.

## References

1. N.D. Ahakarchy, Z.N. Abdullah, Z.M. Alameen, Z.A. Harjan, Audio verification in forensic investigation using light deep neural network. *Int. J. Inf. Technol.* **16**(5), 2813–2821 (2024)
2. B.S. Atal, The history of linear prediction. *IEEE Signal Process. Mag.* **23**(2), 154–161 (2006)
3. Z. Bai, X. Zhong, Speaker recognition based on deep learning: an overview. *Neural Netw.* **140**, 65–99 (2021)
4. G. Baldini, I. Amerini, C. Gentile, Microphone identification using convolutional neural networks. *IEEE Sens. Lett.* **3**(7), 1–4 (2019)
5. R. Buchholz, C. Kraetzer, J. Dittmann, Microphone classification using Fourier coefficients, in *Proceedings of Information Hiding, 11th International Workshop*, pp. 235–246 (2009)
6. F. Busquet, F. Efthymiou, C. Hildebrand, Voice analytics in the wild: validity and predictive accuracy of common audio-recording devices. *Behav. Res. Methods* **56**(3), 2114–2134 (2024)
7. W.M. Campbell, Generalized linear discriminant sequence kernels for speaker recognition, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1, pp. 161–164 (2002)
8. R. Chakroun, M. Frikha, A deep learning approach for text-independent speaker recognition with short utterances. *Multimed. Tools Appl.* **82**, 1–23 (2023)
9. Z. Chen, M. Lin, Z. Wang, Q. Zheng, C. Liu, Spatio-temporal representation learning enhanced speech emotion recognition with multi-head attention mechanisms. *Knowl. Based Syst.* **281**, 111077 (2023)
10. N. Dehak, P.J. Kenny, R. Dehak, P. Dumouchel, P. Ouellet, Front-end factor analysis for speaker verification. *IEEE Trans. Audio Speech Lang. Process.* **19**, 788–798 (2011)
11. M. Geng, X. Xie, Z. Ye, T. Wang, G. Li, S. Hu, X. Liu, H. Meng, Speaker adaptation using spectro-temporal deep features for dysarthric and elderly speech recognition. *IEEE/ACM Trans. Audio Speech Lang. Process.* **30**, 2597–2611 (2022)
12. C. Haniłçi, F. Ertas, Optimizing acoustic features for source cell-phone recognition using speech signals, in *Proceedings of the First ACM Workshop on Information Hiding and Multimedia Security*, pp. 141–148 (2013)
13. C. Haniłçi, F. Ertas, T. Ertas, Ö. Eskidere, Recognition of brand and models of cell-phones from recorded speech signals. *IEEE Trans. Inf. Forensics Secur.* **7**(2), 625–634 (2012)
14. M. Hariharan, L.S. Chee, S. Yaacob, Analysis of infant cry through weighted linear prediction cepstral coefficients and probabilistic neural network. *J. Med. Syst.* **36**, 1309–1315 (2012)
15. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778 (2016)
16. H. Hermansky, Perceptual linear predictive (plp) analysis of speech. *J. Acoust. Soc. Am.* **87**(4), 1738–1752 (1990)
17. J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7132–7141 (2018)
18. Y.A. Ibrahim, J.C. Odiketa, T.S. Ibiyemi, Preprocessing technique in automatic speech recognition for human computer interaction: an overview. *Ann. Comput. Sci. Ser.* **15**(1), 186–191 (2017)
19. M.M. Kabir, M.F. Mridha, J. Shin, I. Jahan, A.Q. Ohi, A survey of speaker recognition: fundamental theories, recognition methods and opportunities. *IEEE Access.* **9**, 79236–79263 (2021)
20. T. Kinnunen, H. Li, An overview of text-independent speaker recognition: from features to supervectors. *Speech Commun.* **52**(1), 12–40 (2010)
21. C. Kotropoulos, Source phone identification using sketches of features. *IET Biom.* **3**(2), 75–83 (2014)
22. C. Kotropoulos, S. Samaras, Mobile phone identification using recorded speech signals, in *Proceedings of 19th International Conference on Digital Signal Processing*, pp. 586–591 (2014)
23. Y. Lei, N. Scheffer, L. Ferrer, M. McLaren, A novel scheme for speaker recognition using a phonetically-aware deep neural network, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1695–1699 (2014)
24. B. Logan, Mel frequency cepstral coefficients for music modeling, in *Proceedings of Ismir*, 1, pp. 11 (2000)
25. D. Luo, P. Korus, J. Huang, Band energy difference for source attribution in audio forensics. *IEEE Trans. Inf. Forensics Secur.* **13**, 2179–2189 (2018)
26. A.Q. Ohi, M.F. Mridha, M.A. Hamid, M.M. Monowar, Deep speaker recognition: process, progress, and challenges. *IEEE Access.* **9**, 89619–89643 (2021)

27. S.J. Pan, Q. Yang, A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **22**(10), 1345–1359 (2010)
28. Y. Panagakis, C. Kotropoulos, Automatic telephone handset identification by sparse representation of random spectral features, in *Proceedings of the on Multimedia and Security*, pp. 91–96 (2012)
29. Y. Panagakis, C. Kotropoulos, Telephone handset identification by feature selection and sparse representations, in *Proceedings of IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 73–78 (2012)
30. W. Rao, M.W. Mak, Boosting the performance of i-vector based speaker verification via utterance partitioning. *IEEE Trans. Audio Speech Lang. Process.* **21**(5), 1012–1022 (2013)
31. D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, S. Khudanpur, X-vectors: Robust dnn embeddings for speaker recognition, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5329–5333 (2018)
32. T. Suchitha, A. Bindu, Feature extraction using mfcc and classification using gmm. *Int. J. Sci. Res. Dev.* **3**(5), 1278–1283 (2015)
33. E. Variani, X. Lei, E. McDermott, I.L. Moreno, J. Gonzalez-Dominguez, Deep neural networks for small footprint text-dependent speaker verification, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4052–4056 (2014)
34. A. Veit, M.J. Wilber, S. Belongie, Residual networks behave like ensembles of relatively shallow networks, in *Advances in Neural Information Processing Systems*, vol. 29 (2016)
35. Z. Wang, Y. Yang, C. Zeng, S. Kong, S. Feng, N. Zhao, Shallow and deep feature fusion for digital audio tampering detection. *EURASIP J. Adv. Signal Process.* **2022**(69), 1–20 (2022)
36. Z. Wang, J. Zhan, G. Zhang, D. Ouyang, H. Guo, An end-to-end transfer learning framework of source recording device identification for audio sustainable security. *Sustainability* **15**(14), 11272 (2023)
37. C. Zeng, S. Feng, Z. Wang, X. Wan, Y. Chen, N. Zhao, Spatio-temporal representation learning enhanced source cell-phone recognition from speech recordings. *J. Inf. Secur. Appl.* **80**, 103672 (2024)
38. C. Zeng, S. Feng, Z. Wang, Y. Zhao, K. Li, X. Wan, Audio source recording device recognition based on representation learning of sequential gaussian mean matrix. *Forensic Sci. Int. Digit. Investig.* **48**, 301676 (2024)
39. C. Zeng, S. Feng, D. Zhu, Z. Wang, Source acquisition device identification from recorded audio based on spatiotemporal representation learning with multi-attention mechanisms. *Entropy* **25**(4), 626 (2023)
40. C. Zeng, S. Kong, Z. Wang, S. Feng, N. Zhao, J. Wang, Deletion and insertion tampering detection for speech authentication based on fluctuating super vector of electrical network frequency. *Speech Commun.* **158**, 103046 (2024)
41. C. Zeng, S. Kong, Z. Wang, K. Li, Y. Zhao, Digital audio tampering detection based on deep temporal-spatial features of electrical network frequency. *Information* **14**(5), 253 (2023)
42. C. Zeng, S. Kong, Z. Wang, K. Li, Y. Zhao, X. Wan, Y. Chen, Digital audio tampering detection based on spatio-temporal representation learning of electrical network frequency. *Multimed. Tools Appl.* **2024**, 1–23 (2024)
43. C. Zeng, K. Li, Z. Wang, Enformer: long-short term representation of electric network frequency for digital audio tampering detection. *Knowl. Based Syst.* **297**, 111938 (2024)
44. C. Zeng, Y. Yang, Z. Wang, S. Kong, S. Feng, Audio tampering forensics based on representation learning of enf phase sequence. *Int. J. Digit. Crime Forensics* **14**(1), 1–19 (2022)
45. C. Zeng, D. Zhu, Z. Wang, M. Wu, W. Xiong, N. Zhao, Spatial and temporal learning representation for end-to-end recording device identification. *EURASIP J. Adv. Signal Process.* **2021**(1), 1–19 (2021)
46. C. Zeng, D. Zhu, Z. Wang, Z. Wang, N. Zhao, L. He, An end-to-end deep source recording device identification system for web media forensics. *Int. J. Web Inf. Syst.* **16**(4), 413–425 (2020)
47. Q. Zheng, Z. Chen, Z. Wang, H. Liu, M. Lin, Meconformer: highly representative embedding extractor for speaker verification via incorporating selective convolution into deep speaker encoder. *Expert Syst. Appl.* **244**, 123004 (2024)
48. L. Zou, Q. He, X. Feng, Cell phone verification from speech recordings using sparse representation, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1787–1791 (2015)
49. L. Zou, Q. He, J. Wu, Source cell phone verification from speech recordings using sparse representation. *Digit. Signal Process.* **62**, 125–136 (2017)
50. L. Zou, Q. He, J. Yang, Y. Li, Source cell phone matching from speech recordings by sparse representation and kiss metric, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2079–2083 (2016)



**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.