Audio Classification System Based on One-Dimensional Convolutional Neural Networks

Ruiqing Li, Tianyuan Liu
School of Artificial Intelligence and Big Data, Henan University of Technology, Zhengzhou, Henan, China

Abstract: Audio classification holds significant practical value in fields such as intelligent security, healthcare, autonomous driving, and smart education. This study implements an audio classification system based on a one-dimensional convolutional neural network (1D-CNN). The system employs 1D-CNN for audio classification and introduces a progressive Dropout strategy to address issues of model overfitting and poor generalization capability. The system adopts a modular design comprising five major components: data preprocessing, model training, model evaluation, prediction, and result visualization. During data preprocessing, audio data is collected, cleaned, formatted, and processed to extract Mel Frequency Cepstral Coefficients (MFCC) as features. Data augmentation techniques are simultaneously applied to enhance model generalization. For model construction, a 1D-CNN forms the foundational convolutional layer, complemented by batch normalization, max-pooling, Dropout, and fully-connected layers for feature extraction and classification. Training employs the Adam optimizer with a dynamic learning rate scheduling mechanism. Experimental results demonstrate the system achieves 78.4% accuracy and 84.1% MAP@3 on a 21-category audio dataset, validating the model's effectiveness.

Keywords: Audio Classification; 1D-CNN; MFCC; Deep Learning; Dropout

1. Introduction

Audio classification possesses substantial practical significance, finding extensive applications in intelligent security, healthcare, autonomous driving, smart education, and smart home sectors. Traditional audio classification typically utilizes feature extraction and classification, template matching, and threshold-based methods. For feature extraction, manual techniques based on linear frequency analysis-such as Mel Frequency Cepstral Coefficients (MFCC) [1] and other manual feature extraction methods-are employed. These features are subsequently processed by classifiers such as Support Vector Machines (SVM) [2]. While these methods have played a crucial role in early audio processing, their limitations have become evident as audio data becomes increasingly complex and diverse. These limitations encompass restricted model generalization capabilities, difficulties in handling intricate audio signals, and challenges in model optimization.

In recent years, with the advancement of deep learning technology, researchers have commenced the application of deep learning models to audio classification, including Convolutional Neural Networks (CNN) [3], Recurrent Neural Networks (RNN) [4], and Convolutional Recurrent Neural Networks (CRNN) [5]. These methodologies have markedly enhanced feature extraction and model generalization capabilities, facilitating improved management of intricate audio classification tasks. However, while CNNs excel in local feature extraction, they encounter difficulties in addressing long-term dependencies; RNNs can capture sequential information but are susceptible to gradient vanishing or exploding during training; and CRNNs amalgamate the strengths of both yet present complex structures and challenging training processes. This research utilizes a one-dimensional convolutional neural network (1D-CNN) to develop an audio classification system, integrating a progressive dropout strategy [6]. This method ensures structural simplicity while significantly improving model accuracy and generalization performance.

2. System Architecture and Execution

This system implements a deep learning framework for audio classification. The system employs a modular design consisting of five interrelated modules: data preprocessing, model training, model evaluation, prediction, and result visualization. These modules cooperate to guarantee smooth and effective workflow execution. The system is architecturally constructed with Python and the Flask framework, utilizing librosa for audio feature extraction and implementing a 1D-CNN model for classification predictions. Figures 1 and 2 depict the audio categorization processing flow and the interface of the system's front-end.

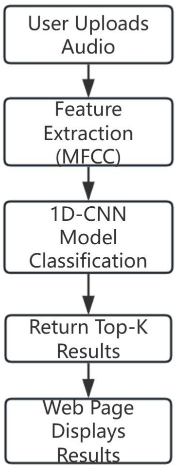
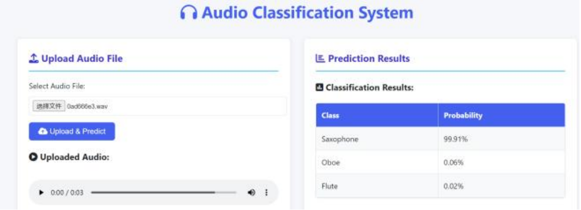

User Uploads Audio
Feature Extraction (MFCC)
1D-CNN Model Classification
Return Top-K Results
Web Page Displays Results

Figure 1. Audio Classification Processing Flowchart



Class	Probability
Saxophone	99.91%
Oboe	0.06%
Flute	0.02%

Figure 2. Front-end Interface

3. Data Preprocessing

Data preprocessing is a critical step in audio classification systems, directly impacting feature quality and model performance. This chapter details three key steps: data preparation, feature extraction, and data augmentation.

3.1 Data Preparation

The data preparation phase involves collecting, cleaning, and formatting raw audio data to ensure data quality and consistency. The dataset used in this study is provided by the "Automatic Audio Labeling Challenge" of the iFlytek AI Developer Competition. This dataset contains 21 audio categories (13 musical instruments and 8 environmental sound effects), totaling 5,637 annotated samples. First, preliminary cleaning was performed to remove data with invalid paths or audio reading errors. Subsequently, the raw audio was formatted and features such as MFCCs were extracted. To facilitate model training, audio file labels were numerically encoded, mapping the 21 audio categories to integers ranging from 0 to 20. Furthermore, a serialization mechanism ensured consistency in encoding states, laying the foundation for subsequent feature extraction and model training.

3.2 MFCC Feature Extraction

MFCC feature extraction is the core component of building a speech recognition system. This study utilizes the built-in functions of the librosa library to extract Mel Frequency Cepstral Coefficients (MFCC). The MFCC feature extraction process is designed to approximate the perceptual characteristics of the human auditory system.Through carefully selected parameter configurations, it effectively captures the spectral features of the audio signal. The specific steps are as follows:

(1) Pre-emphasis: Pre-emphasis involves applying a filter to the speech signal to boost its high-frequency components, resulting in a flatter spectral response. This system employs a first-order high-pass filter with the transfer function.

$$y(n) = x(n) - 0.97x(n-1) \tag{1}$$

Where $x(n)$ denotes the nth sample of a discrete-time signal.

(2) Framing and Window Application: An overlapping region has been added between subsequent frames in continuous speech in order to reduce excessive variation. Framing is thus required for obtaining "short-term" signals. However, after framing, discontinuities may arise between frames due to the reduction of overlapping areas. Therefore, windowing is necessary. The used Hamming window function is:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) \tag{2}$$

Here, N denotes the frame size.

(3) Feature Extraction: First, the windowed signal undergoes a Fourier transform to obtain its frequency domain representation. Next, the energy distribution across each frequency band

is calculated. Finally, the discrete cosine transform converts the frequency domain energy into feature coefficients using the following formula:

$$c_m = \sum_{k=1}^{K} log\,(E_k)\,cos\,[\frac{\pi m}{K}(k - 0.5)] \quad (3)$$

Where C_m is the m th transform coefficient. E_k is the energy of the kth frequency band. K is the number of frequency bands. m is the index of the transform coefficient, ranging from 0 to $K - 1$. This study employs a 40-dimensional feature space to enhance the capture of spectral details. Feature post-processing utilizes time-averaged pooling to convert variable-length audio sequences into fixed-dimensional feature vectors, ensuring computational efficiency while improving feature stability.

3.3 Data Augmentation

To enhance the model's generalization capability in complex acoustic environments, this study employs audio data augmentation strategies [7]. Pitch-shift augmentation is based on the phase vocoder principle, where

$$\Delta f = \pm 2 \quad (4)$$

a semitone offset to simulate individual performer variations. Time stretching enhancement employs an improved time-domain processing algorithm, where the stretching coefficient

$$\rho = 0.8 \quad (5)$$

Modify audio duration without altering pitch characteristics. The enhancement parameters have been thoroughly validated to ensure the preservation of class discriminative features while introducing data diversity. Through the aforementioned data augmentation techniques, the dataset has been effectively expanded, enhancing the model's robustness and generalization capabilities, thereby laying a solid foundation for subsequent model training and evaluation.

4. Model Construction

4.1 Model Architecture Design

This study employs a one-dimensional convolutional neural network (1D-CNN) architecture for audio classification tasks. This architecture is particularly well-suited for processing sequential data such as audio signals. The model's network architecture diagram is shown in Figure 3.

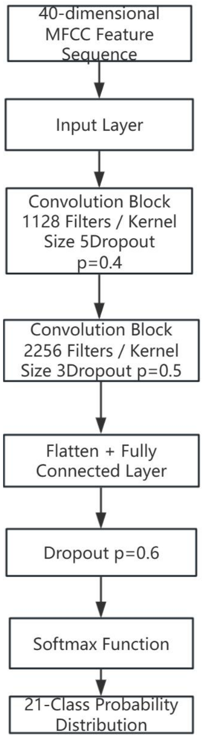

40-dimensional MFCC Feature Sequence
Input Layer
Convolution Block 1128 Filters / Kernel Size 5Dropout p=0.4
Convolution Block 2256 Filters / Kernel Size 3Dropout p=0.5
Flatten + Fully Connected Layer
Dropout p=0.6
Softmax Function
21-Class Probability Distribution

Figure 3. Model Architecture Diagram

To better understand how the model operates, each layer will be introduced step by step below.

(1) Input Layer The input layer first processes the 40-dimensional MFCC features, transforming them into a tensor of dimension (40, 1) to provide a suitable input format for subsequent convolutional operations.

(2) Convolution Layer

This study employs two 1D-CNN convolutional blocks. The first block uses 128 filters of size 5, while the second block uses 256 filters of size 3. Both blocks utilize the ReLU activation function [8], followed by batch normalization and max pooling layers.

The convolution operation can be expressed as:

$$C_{i,j} = ReLU(\sum_k W_{i,k} \cdot X_{k,j} + b_i) \quad (6)$$

Where $C_{i,j}$ is the output of the i th filter at position j, $W_{i,k}$ is the filter weight. $X_{k,j}$ is the input data,and b_i is the bias term. The convolutional operation is performed between the input data and the filter weights. After adding the bias term and passing through the *ReLU* activation function. Batch normalization and max pooling are applied to obtain the final output feature map.

(3) Dropout Layer

To prevent overfitting, this study introduces Dropout layers after each convolutional layer and fully connected layer. The Dropout mechanism employs an incremental configuration, which can be understood through the following formula:

$$h_i = r_i \cdot h_i,\ where\ r_i \sim Bernoulli(p) \quad (7)$$

Where h_i is the output or hidden layer activation value of thei th neuron in the neural network, r_i is a binary random variable (0 or 1), $Bernoulli(p)$ is r_i, which follows a Bernoulli distribution with parameterp , andp is the probability that the neuron is retained. The p-value for the Dropout layer after the first convolutional block is set to 0.4, the p-value for the second Dropout layer is 0.5, and the p-value for the second Dropout layer is increased to 0.6. This strategy enhances the model's feature learning capability while effectively improving its generalization performance.

(4) Fully Connected Layer and Output Layer

After feature extraction, the feature maps are flattened into one-dimensional feature vectors via a Flatten layer and fed into the fully connected layer. The nonlinear transformation in this layer is represented as:

$$h = \mathrm{Re}\,LU(w_x + b) \quad (8)$$

The $ReLU$ activation function is employed to fuse and abstract high-level acoustic features. Here, h represents the hidden layer output, W denotes the weight matrix, x is the input feature vector, and b is the bias term.

Finally, the output layer uses the Softmax function [9] to compute the category probability distribution:

$$\mathrm{p_{i,c}} = \frac{\mathrm{e}^{z_{i,c}}}{\sum_{j=1}^{c} e^{z_{i,j}}} \quad (9)$$

Here, $\mathrm{p}_{i,c}$ is the predicted probability for the$i$ th category. Since the dataset has 21 categories, $i = 1,2,...,21$ provides reliable decision support for multi-class classification tasks. C also represents the number of categories, $z_{i,c}$ denotes the raw output for thei th category, $\mathrm{e}^{z_{i,c}}$ is the exponential function, $\sum_{\mathrm{j=1}}^{21} e^{z_{i,j}}$ and is the sum of exponentials for all categories.

4.2 Model Optimization Strategy

Model training employs the Adam optimizer [10], utilizing the sparse classification cross-entropy loss function [11] as the objective function to guide the Adam optimizer in adjusting model parameters. The sparse classification cross-entropy loss function is defined as:

$$\mathrm{L} = -\sum_{\mathrm{i=1}}^{N} \ln(p_{i,y_i}) \quad (10)$$

L denotes the total loss function value, N represents the total number of training samples, y_i is the true category label of the i-th sample (an integer such as 0, 1, 2, ..., 20), and p_{i,y_i} is the predicted probability that the i-th sample belongs to the true category $\mathrm{y_i}$. Simultaneously, the ReduceLROnPlateau dynamic learning rate scheduling mechanism [12] is introduced, which automatically reduces the learning rate during validation loss plateaus to promote fine-grained model convergence. An early stopping strategy is also employed during training, continuously monitoring validation set performance to ensure model training terminates at optimal generalization.

5. Results Analysis

5.1 Experimental Setup

Accuracy and MAP@K [13] are selected as core evaluation metrics. Accuracy measures overall classification effectiveness, while MAP@K (Mean Average Precision) specifically evaluates the average precision among the top K predictions. Its formula is:

$$\mathrm{MA}P@K = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\min(m,K)}\sum_{k=1}^{K} P_i(k)\cdot rel_i(k) \quad (11)$$

Where $P_i(k)$ denotes the precision of the top k predictions, and $rel_i(k)$ is the relevance indicator function. This metric effectively reflects the model's practical value in multi-candidate predictions. In this experiment, $K=3$ is set to evaluate Top-3 prediction performance.

5.2 Model Performance Analysis

The 1D-CNN model constructed in this study achieved good performance on the validation set. As shown in Table 1, it attained an accuracy of 78.4% and a MAP@3 score of 84.1%.Observing Figure 4 of the training process, both training and validation accuracy steadily increased with training iterations, ultimately approaching 85%, indicating effective model learning. Concurrently, training and validation losses continuously decreased and stabilized, demonstrating consistent model performance across both training and validation sets.

Table 1. Model Evaluation Summary

Evaluation Metric	Value
Training Accuracy	78.4%
Validation Accuracy	82.6%
MAP@3	84.1%

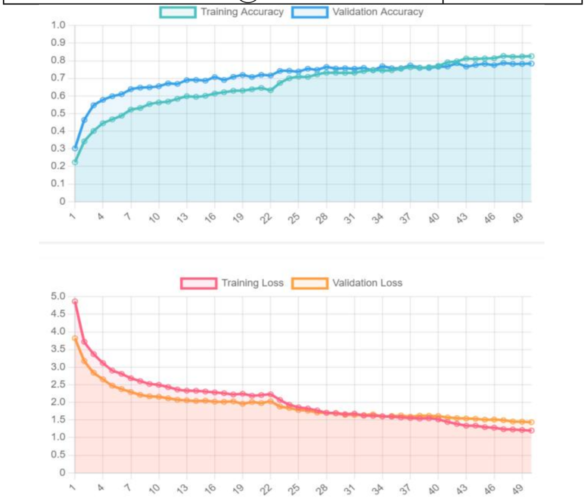


Figure 4. Training and Validation Performance Plot

5.3 Analysis of Recognition Performance across Categories

To gain deeper insights into the model's performance, we further analyzed its behavior across different categories. As shown in Figure 5 (confusion matrix) and Table 2 (category-specific classification performance), the model performs well in most categories, particularly in "Oboe" and "Writing," where Top-1 accuracy reaches 98.3% and 85.2%, respectively, indicating strong recognition capabilities in these categories. However, the Top-1 accuracy for the "Electric_piano" and "Trumpet" categories is relatively low at 46.7% and 71.7%, respectively. Additionally, the confusion matrix shows a significant number of misclassifications, indicating that the model still struggles to distinguish between these instrument sounds and exhibits some misclassification. This may be due to acoustic similarities between the two instruments or the features used being insufficiently sensitive for distinguishing them. Overall, the model demonstrates strong performance in Top-3 accuracy, exceeding 80% for most categories.

5.4 Performance Comparison of Different Dropout Methods

To validate the effectiveness of progressive Dropout, this study examined model accuracy and overfitting levels under different Dropout configurations, as shown in Table 3.

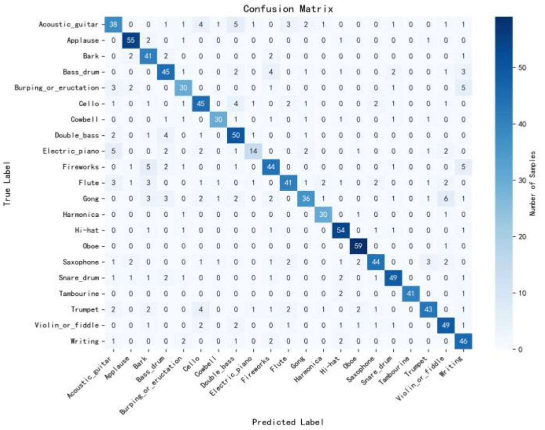


Figure 5. Confusion Matrix

Table 2. Classification Performance by Category

Category	Top-1 Accuracy	Top-3 Accuracy
Acoustic_guitar	63.3%	90.0%
Applause	91.7%	93.3%
Bark	85.4%	89.6%
Bass_drum	75.0%	91.7%
Burping or Eructation	71.4%	81.0%
Cello	75.0%	95.0%
Cowbell	78.9%	78.9%
Double Bass	83.3%	91.7%
Electric_piano	46.7%	83.3%
Fireworks	73.3%	93.3%
Flute	68.3%	91.7%
Gong	61.0%	81.4%
Harmonica	90.9%	93.9%
Hi-hat	90.0%	95.0%
Oboe	98.3%	98.3%
Saxophone	73.3%	90.0%
Snare_drum	81.7%	95.0%
Tambourine	93.2%	97.7%
Trumpet	71.7%	81.7%
Violin or fiddle	81.7%	95.0%
Writing	85.2%	96.3%

Table 3. Performance Comparison under Different Dropout Settings

Dropout Configuration	Validation Set Accuracy	Training Set Accuracy
All 0.5	77.4%	84.5%
All 0.6	75.1%	76.2%
Progressive (0.4-0.5-0.6)	78.4%	82.6%

By comparing model performance across different configurations, when the Dropout rate is uniformly set to 0.5, the model achieves 77.4% accuracy on the validation set and 84.5% on the training set, indicating some overfitting. When the Dropout rate was uniformly set to 0.6, although overfitting was mitigated, the validation set accuracy dropped to 75.1% and the training set accuracy decreased to 76.2%. This suggests that an excessively high Dropout rate may have impaired the model's learning efficiency. In contrast, the progressive Dropout configuration (0.4-0.5-0.6) yielded the highest validation set accuracy at 78.4%, while maintaining training set accuracy at 82.6%. This demonstrates that progressive Dropout better balances model training and generalization capabilities, thereby preventing overfitting while preserving effective learning performance.

6. Conclusions and Future Directions

This paper successfully implements an audio event classification system based on a one-dimensional convolutional neural network (1D-CNN). By introducing progressively increasing dropout rates across layers, the model's generalization capability and accuracy are effectively enhanced. Experimental results demonstrate that the system achieves 78.4% accuracy and 84.1% MAP@3 on a dataset containing 21 audio categories, validating the effectiveness of the optimization strategy. The system holds potential for further refinement: Attention mechanisms could be integrated to enhance the model's focus on critical acoustic features, particularly when distinguishing spectrally similar audio categories. At the data level, richer acoustic transformation enhancement techniques—such as background noise injection and spectral masking—can be expanded to bolster the model's robustness in real-world scenarios. For practical application, the system can be extended into an online system supporting real-time streaming processing and integrated with visualization components to enhance usability and user-friendliness. Through continuous optimization, this system holds promise for greater application value in fields such as environmental awareness, intelligent monitoring, and smart education.

References

[1] Rashed A, Abdulazeem Y, Farrag A T, et al. Toward Inclusive Smart Cities: Sound-Based Vehicle Diagnostics, Emergency Signal Recognition, and Beyond. Machines, 2025, 13(4): 258-258.

[2] Jaganathan S J, Ali M J, Abdullah S R S. Unlocking the potential of artificial intelligence in wastewater treatment: Innovations, opportunities, and challenges. Journal of Environmental Chemical Engineering, 2025, 13(6): 119671-119671.

[3] Cerna D P, Cascaro J R, Juan S O K, et al. Bisayan Dialect Short-time Fourier Transform Audio Recognition System using Convolutional and Recurrent Neural Network. International Journal of Advanced Computer Science and Applications (IJACSA), 2023, 14(3).

[4] Usha M, Priyanka C. Acoustic-Based Emergency Vehicle Detection Using an Ensemble of Deep Learning Models. Procedia Computer Science, 2023, 218227-234.

[5] Anam B, Kumar N G. Robust technique for environmental sound classification using convolutional recurrent neural network. Multimedia Tools and Applications, 2023, 83(18): 54755-54772.

[6] Chen K, Wang A Z. Review of Regularization Methods for Convolutional Neural Networks. Research on Computer Applications, 2024, 41(04): 961-969.

[7] Jaganathan S J, Ali M J, Abdullah S R S. Unlocking the potential of artificial intelligence in wastewater treatment: Innovations, opportunities, and challenges. Journal of Environmental Chemical Engineering, 2025, 13(6): 119671.

[8] Zhang H M, Wu F J, Zheng C, et al. Two Properties of the Nonlinear Activation Function ReLU on the MNIST Dataset. Journal of Hubei Normal University (Natural Science Edition), 2024, 44(03): 1-7.

[9] Nguyen V T. Research on Neural Network Activation Functions for Handwritten Character and Image Recognition. Xidian University, 2020.

[10]Anto W, Napitupulu H, Gusriani N. Edelweiss Flower Species Classification Using Convolutional Neural Network with Adaptive Moment Estimation Optimizer (Adam). IAENG International Journal of Computer Science, 2025, 52(9).

[11]Cui Y X. Cross-Entropy-Based Stochastic Weighted Network. Hebei University,

2017.

[12]Dharanalakota V, Raikar A A, Ghosh K P. Improving neural network training using dynamic learning rate schedule for PINNs and image classification. Machine Learning with Applications, 2025, 21100697-100697.

[13]Hamid H, Javad M E, Azadeh M. LVTIA: A new method for keyphrase extraction from scientific video lectures. Information Processing and Management, 2022, 59(2).