Study on the Application of Improved Audio Recognition Technology Based on Deep Learning in Vocal Music Teaching

Nan Liu

Shan Dong College of Arts, Shan Dong 250000, Jinan, China

Correspondence should be addressed to Nan Liu; z00834@sdca.edu.cn

Received 27 June 2022; Revised 21 July 2022; Accepted 26 July 2022; Published 18 August 2022

Academic Editor: Wenlong Hang

Copyright © 2022 Nan Liu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As one of the hotspots in music information extraction research, music recognition has received extensive attention from scholars in recent years. Most of the current research methods are based on traditional signal processing methods, and there is still a lot of room for improvement in recognition accuracy and recognition efficiency. There are few research studies on music recognition based on deep neural networks. This paper expounds on the basic principles of deep learning and the basic structure and training methods of neural networks. For two kinds of commonly used deep networks, convolutional neural network and recurrent neural network, their typical structures, training methods, advantages, and disadvantages are analyzed. At the same time, a variety of platforms and tools for training deep neural networks are introduced, and their advantages and disadvantages are compared. TensorFlow and Keras frameworks are selected from them, and the practice related to neural network research is carried out. Training lays the foundation. Results show that through the development and experimental demonstration of the prototype system, as well as the comparison with other researchers in the field of humming recognition, it is proved that the deep-learning method can be applied to the humming recognition problem, which can effectively improve the accuracy of humming recognition and improve the recognition time. A convolutional recurrent neural network is designed and implemented, combining the local feature extraction of the convolutional layer and the ability of the recurrent layer to summarize the sequence features, to learn the features of the humming signal, so as to obtain audio features with a higher degree of abstraction and complexity and improve the performance of the humming signal. The ability of neural networks to learn the features of audio signals lays the foundation for an efficient and accurate humming recognition process.

1. Introduction

Traditional text-based retrieval techniques are still widely used in the field of music retrieval. There are many problems to be solved in music retrieval based on text information [1]. Firstly, users need to know the name, singer, music style and other information of the song they are looking for. Without this information, they cannot find the music they are interested in. Secondly, in the text-based music retrieval system, the music in the music library needs to be associated with various additional information. This kind of labeling work is difficult to be completed automatically by machines, requiring a lot of manpower and high cost [2].

Researching more efficient music retrieval technology is a work with great practical value. Content-based music recognition is a research hotspot in the field of music retrieval in recent years. Compared with using text retrieval methods, content-based music retrieval is more convenient. The music retrieval based on humming recognition can also be well combined with the traditional text retrieval technology to provide a more accurate and richer music retrieval method. Humming recognition-related technologies have broad application prospects and are useful in practical applications. At the same time, there are still few research methods for humming recognition based on deep learning, and there is still a lot of research space for humming recognition models based on deep learning. Therefore, this subject has good research prospects and is a research subject worthy of in-depth exploration [3].
Humming recognition has the advantages of user-friendly interaction and convenient use on mobile devices. The main technologies used in humming recognition research can be summarized into the following types [4]: recognition technology based on symbol matching, recognition technology based on melody matching, and recognition technology based on statistical models. The recognition technology based on symbol matching is developed from the traditional string matching method. It generally first extracts the note information to obtain the note sequence. Then the note sequence is regarded as a string, and the similarity of the note sequence is obtained by using the string matching correlation algorithm, and the recognition result is obtained accordingly [5]. The recognition technology based on melody matching first extracts the humming audio pitch signal, connects the curves of pitch changing with time to form a melody curve, and then analyzes and matches the melody feature with the audio melody feature in the database to obtain the recognition result. The recognition technology based on a statistical model utilizes the time-domain features or frequency-domain features of audio, adopts statistical models such as Hidden Markov, etc. to model the songs in the database, and then calculates the probability of humming audio predicted by the model, with the maximum probability. The songs are returned as the recognition result [3].

In 1995, Ghias et al. developed and studied the first humming recognition system, which uses a typical string matching-based recognition technology, and uses the letters $U$ (up), $D$ (down), or $S$ for the pitch change of the audio signal (unchanged) to represent the humming audio signal using a string consisting of these three characters, and then use the string matching algorithm to calculate the matching probability of the song in the database [6]. The research from McNaughton et al. is also based on symbol matching technology [7]. They extracted the rhythm information and pitch change information of music, used the form of strings to represent such audio features, and verified the effectiveness of the method in the humming recognition system through experiments. In 1999, Kosugi et al. proposed a method to measure similarity by using Euclidean distance based on audio pitch and rhythm information. Scholars such as Calissie proposed a new method in 2002, which introduced an auditory model and achieved good recognition accuracy improvement on public datasets [8]. Shih et al. creatively introduced the hidden Markov model in their research, using the audio pitch information as the input of the HMM, thus proving the feasibility of using the statistical model for humming recognition research [9]. Downie et al. introduced a dynamic time scaling algorithm for the robustness of the humming recognition system, which greatly improved the overall fault tolerance of the humming recognition system [10]. Pardo et al. comprehensively adopted a matching algorithm based on the hidden Markov model and a distance matching algorithm for similarity measurement in humming recognition [11].

There are two main problems with the current humming recognition technology. The first is that the recognition accuracy still needs to be improved, especially when there is a partial deviation in the user’s humming, it is difficult for the existing music feature-based methods to obtain a satisfactory recognition accuracy [12]. On the other hand, there is a problem that the processing time is too long when using the feature matching correlation algorithm for humming recognition [13]. Extracting and processing relevant features from the user’s humming and matching with the songs in the database require a certain computing time, which brings a long waiting time to the user interaction process and is not user-friendly.

Deep learning is a new research direction in the field of machine learning [14]. It is mainly based on artificial neural networks and uses multilayer representations to model complex relationships between data. In addition to the ability of traditional machine learning methods to discover the relationship between data features and tasks, deep learning can also summarize more abstract and complex features from simple feature learning [15]. In recent years, breakthrough achievements have been made in the fields of computer vision, image processing, natural language understanding, etc., which have attracted widespread attention.

The development of deep learning can be roughly divided into three stages. Early neural network models were similar to bionic machine learning, which tried to mimic the learning mechanism of the brain [16]. The earliest neural network mathematical model was proposed by Professor Warren and Walter in 1943. In order to allow the computer to set the weights more automatically and reasonably, Professor Frank Rosenblatt proposed the perceptron model in 1958. The perceptron is the first model that can learn feature weights based on sample data [14]. However, under the limitation of computing power at that time, these research results were not taken seriously. At the end of the 1980s, the second wave of neural network research climax came with the proposal of distributed knowledge representation and neural network back-propagation algorithm. Neural network research in this period used multiple neurons to express knowledge and concepts in the real world, which greatly enhanced the expressive ability of the model and laid the foundation for later deep learning [17].

The third high-speed development stage of neural network research comes with the improvement of computer performance and the development of cloud computing, GPU, and other technologies [18]. With the solid foundation provided by these hardware resources, the amount of computation is no longer an issue that hinders the development of neural networks. At the same time, with the popularization of the Internet and the development of search technology, people can easily obtain a large amount of data and information, which solves the long-standing problem of missing datasets in neural network training [19]. At this stage, deep learning has truly ushered in a development climax and has repeatedly achieved breakthrough results in many fields.

In the field of speech recognition, the traditional GMM-HMM speech recognition model has encountered development bottlenecks after years of research, and the introduction of deep-learning-related technologies has significantly improved the accuracy of speech recognition [20]. Since the concept of deep learning was introduced into the field of speech recognition in 2009, in just a few years, the
2. Methods

2.1. Audio Signal Processing Flow. First of all, for the test humming data, a digitized audio signal needs to be obtained through a process of sampling and quantization. Then, it is necessary to perform certain preprocessing on the original humming data and test data, including filtering, pre-emphasis, windowing, and framing, to reduce the interference of audio signal noise and improve the saliency of features. Secondly, in the process of training and recognition, it is also necessary to detect the starting point of the note. From the starting point of the note, the data set is intercepted to eliminate the interference of silent segment noise and improve the validity of the data, and the audio signal processing flow is shown in Figure 1.

2.2. Sampling and Quantization. An important step in converting the original humming signal into a digital signal is sampling and quantization. After sampling and quantization, the analog signal becomes a digital audio signal that is discrete in time and amplitude. The sampling theorem points out that a necessary condition that needs to be satisfied in the sampling process is to use a sampling frequency that is greater than twice the bandwidth of the audio signal so that the sampling operation will not lose the information of the original audio signal, and it can also be restored from the sampled digital signal [27]. For human voice signals, the frequency spectrum of the voiced signal is mainly concentrated in the low-frequency band below 4 kHz, and the frequency spectrum of the unvoiced signal is very wide, extending to the high-frequency band above 10 kHz. In the research of this paper, the sampling frequency of 16 kHz is uniformly used for the sampling of hummin audio, so as to ensure that the humming information will not be lost.

After sampling, the signal needs to be quantized. The quantization process can convert the continuous amplitude value of the signal on the time axis into discrete amplitude values [28]. The error generated in the quantization process is called quantization noise, which can be obtained by calculating the difference between the quantized discrete amplitude value and the original signal. In general, quantization noise has the following characteristics (1) it is a stationary white noise; (2) the quantization noise is irrelevant to the input signal; (3) the quantization noise is uniformly distributed within the quantization interval, that is, it has the characteristic of equal probability density distribution. The power ratio between the original audio signal and the quantization noise is called the quantization signal-to-noise ratio and is often used to characterize audio quality. In general, the amplitude of the speech signal obeys the Laplace distribution, and the quantized signal-to-noise ratio can be expressed by

\[ S = 6.02B - 7.2, \]  

where \( S \) is the signal-to-noise ratio and \( B \) is the quantized word length. (1) shows that the word length of each bit in the quantizer corresponds to a quantized signal-to-noise ratio of about 6 dB. When the quantized signal-to-noise ratio reaches 35 dB and above, the audio quality can meet the requirements of general communication systems, so generally, the quantization word length should be greater than 7 dB bits. In practical applications, a word length of more than 12 bits is often used for quantization, because the variation range of the speech waveform can reach up to...
55 dB. In order to maintain a signal-to-noise ratio of 35 dB within this range, an additional 9 bit word length is used for compensation. The dynamic range of the speech waveform around 30 dB changes.

2.3. Humming Signal Preprocessing. For the humming signal input from the recording, quantization noise will be generated when it is converted from quantization to digitization, and there will also be power frequency interference, aliasing interference, etc. In order to reduce these noises, the analysis and feature parameter extraction of the humming signal requires the interference generated, first of all, to filter the humming signal to be processed [29].

The prefiltering operation first detects the frequency of each frequency domain component in the input signal and suppresses the components whose frequency value exceeds half of the sampling frequency to prevent aliasing interference. Then, suppress the power frequency interference of about 50 Hz. In the experiment of this paper, a bandpass filter is used to prefilter the humming audio, the upper cut-off frequency is set to 3400 Hz, and the lower cut-off frequency is set to 60 Hz to filter out the interference of the power frequency.

The humming signal is easily affected by two types of noises: glottal excitation and mouth-nose radiation, and its effective components are relatively small in the high-frequency part. When finding the spectrum of the speech signal, the high-frequency part is more difficult to obtain than the low-frequency part. For the high-frequency part, additional processing is required, that is, pre-emphasis of the humming signal is performed first to increase the proportion of the high-frequency part of the humming signal, and smooth the spectrum of the humming signal, improving the high-frequency resolution of the humming part, which facilitates audio analysis operations in the frequency domain [30].

The pre-emphasis process is generally completed by a pre-emphasis digital filter after the audio signal digitization operation is completed. This type of filter has the ability to improve high-frequency characteristics. First-order digital filters are often used. After pre-emphasis processing, the audio signal can be expressed as

\[ H = 1 - \epsilon X^{-1}, \]  

where \( X \) is the incoming humming signal. \( \epsilon \) is weight and its value is regarded as 0.94 in our study.

The humming audio sequence is a one-dimensional signal on the time axis. In order to analyze the signal, the audio signal needs to be regarded as a stable state in a short time of millisecond level. On this basis, the audio signal is windowed and divided into frames to maintain the short-term stationary characteristics of the speech signal, so that the subsequent speech feature vector representation can be performed on a short-term frame, so as to obtain the feature vector time series.

There are generally two segmentation methods for windowing and framing operations, continuous segmentation, and overlapping segmentation [31]. The method of continuous segmentation means that there is no overlap between frames so that discrete frame feature vectors that do not interfere with each other can be obtained. In this study, in order to make a smooth transition between frames and maintain the continuity of features, the method of overlapping segmentation is adopted. In the overlap segment, the overlap of the previous frame and the next frame is usually taken as about 1/2 of the frame length. Specifically, a moving window of finite length can be used to weight the audio signal to achieve framing.

Commonly used window functions include rectangular window, Hamming window, and Hanning window. Their representations are as follows.

Rectangular window:

\[ w(n) = \begin{cases} 
1, & 0 \leq n \leq N - 1, \\
0, & \text{others}. 
\end{cases} \]  

Hamming window:

\[ w(n) = \begin{cases} 
0.54 - 0.46 \cos \left( \frac{2\pi n}{N - 1} \right), & 0 \leq n \leq N - 1, \\
0, & \text{others}. 
\end{cases} \]  

Hanning window:

\[ w(n) = 0.5 \left[ 1 - \cos \left( \frac{2\pi n}{N - 1} \right) \right], \]  

\[ 0, \text{others}. \]
where $N$ is the frame length. The choice of window shape has different effects on the audio signal. Generally speaking, the spectrum obtained by using the rectangular window is smoother, but in the high-frequency part, it is easy to lose waveform details, thus missing some important information; the Hamming window can effectively overcome the problem of information loss in the high-frequency part, but the spectrum is relatively unsmooth; the Hanning window generally needs to use a larger bandwidth, which is about twice the bandwidth of the rectangular window of the same width, and the attenuation effect of the Hanning window is larger than that of the rectangular window, which is much bigger.

The selection of the window length is also very important when adding a window. There is the following relationship between the length of the window $N$, the sampling period $T_s$, and the frequency resolution $\Delta f$:

$$\Delta f = \frac{1}{NT_s} \quad (6)$$

In the case of a certain sampling period, if the window width increases, the frequency resolution will decrease accordingly, so that the details of audio changes cannot be reflected; on the contrary, if the window width $N$ decreases, the frequency resolution increases, and the audio changes become not smooth enough. Therefore, it is necessary to consider the speed of signal change or the level of detail reflected, and set an appropriate window width according to the needs of the head.

Based on the above analysis, this paper uses a Hamming window when windowing and dividing the humming audio signal in the experiment, in which the window length is taken as 5000 points, and the overlap between frames is taken as 2600 points.

2.4. Note Onset Detection. In this paper, a two-threshold method is used for onset detection. The double-threshold method first examines the short-term energy of the humming signal, and the frequency resolution $\Delta f$:

$$\Delta f = \frac{1}{NT_s} \quad (6)$$

Here, $h$ is input signal, $w$ is weight, and $\text{sgn}$ is signal function. The double-threshold method uses a lower threshold $T_1$, and uses this threshold as a reference to obtain the zero-crossing rate of the signal. Generally speaking, the low threshold zero-crossing rate of the noise or silent segment is significantly lower than the high threshold zero-crossing rate of the speech segment, so that the interference of the noise segment can be excluded.

2.5. Humming Signal Feature Representation. Sound is an analog signal, and its one-dimensional time-domain waveform representation can only reflect the relationship of sound pressure with time, but cannot well reflect the characteristics of audio signals. The feature of the audio signal means that the humming signal can be analyzed and processed to remove the irrelevant redundant information in the audio signal and obtain important information that affects the recognition of the humming. Therefore, in order to use the humming audio signal for the deep-learning framework, it is very important to choose a suitable representation of the audio signal features. In the research of speech signal analysis, because the cepstral feature contains more information than other features, it can better characterize the speech signal and is widely used. The commonly used linear prediction cepstral coefficients (Linear Prediction Cepstrum Coefficients, LPCC) and Mel frequency cepstral coefficients (Mel-Frequency Cepstral Coefficients, MFCC) [32].

Since the high-frequency part of the humming signal is easily disturbed by noise, resulting in a frequency shift, most of the effective information in humming recognition is concentrated in the low-frequency part. By converting the linear frequency scale into the Mel frequency scale, the Mel-frequency cepstral coefficient highlights the low-frequency part of the humming signal, that is, it gives information that is more conducive to identification, and at the same time shields the interference of some environmental noises on the audio [33]. Therefore, Mel-frequency cepstral coefficients are more commonly used in most speech and acoustic pattern recognition problems than linear prediction cepstral coefficients. In the study of the humming recognition problem in this paper, it is also used as the input feature vector of the deep-learning framework.

Mel frequency cepstral coefficient (Mel-frequency cepstral coefficient (MFCC)) is a kind of audio signal spectrum feature proposed based on human hearing characteristics, which has a nonlinear correspondence with Hertz frequency. The level of the sound heard by the human ear is not linearly related to the frequency of the sound. The Mel frequency scale is proposed to solve this problem. The use of the Mel frequency scale is in line with the auditory characteristics of the human ear. In the Mel frequency domain, human perception of the pitch has a linear relationship with pitch, that is, if two humming Mel audios are twice as different, then the human perception is also twice as different. Mel-frequency cepstral coefficients can be thought of as folding the short-time Fourier transform on the frequency
axis, reducing the size while preserving the most important perceptible information.

The correlation between Mel frequency \( f_{\text{Mel}} \) and actual \( f \) is as follows:

\[
    f_{\text{Mel}} = 2595 \lg \left( 1 + \frac{f}{700} \right).
\]

The input signal is filtered using a Mel bandpass filter. The effect of each frequency band component is superimposed in the human ear, so the energy in each filter band is superimposed, and then the logarithmic magnitude spectrum of all filters is discretized. Cosine transforms to get Mel frequency cepstral coefficients. The calculation process of Mel frequency is as follows.

Firstly, pre-emphasis, windowing, and framing are performed on the humming signal, and then the frequency spectrum is calculated using the short-time Fourier transform. Secondly, a Mel filter bank of \( L \) channels is set on the Mel frequency, and the \( L \) value is determined by the highest frequency of the signal, generally taking 12-16. Thirdly, pass the linear magnitude spectrum of the signal through the Mel filter to get the filter output \( Y(l) \):

\[
    Y(l) = \sum_{k=0}^{h(l)} w_{f}(k) |x_{f}(k)|, \quad l = 1, 2, \ldots , L,
\]

where \( a(l) \) and \( h(l) \) are the lowest and highest frequencies of the first filter, \( x \) is the input signal, and \( w \) is the weight.

Fourthly, taking the logarithm of the filter output and perform the discrete cosine transform:

\[
    C_{\text{MFCC}}(n) = \sum_{l=1}^{L} \lg Y(l) \times \cos \left( \frac{\pi (l - 0.5) n}{L} \right), \quad n = 1, 2, \ldots , L.
\]

2.6. Experiment Setup. The main hardware used in the experiment is Intel(R) Core(TM) i7-7700K processor and NVIDIA GeForce GTX1070Ti graphics card. In terms of software, the deep-learning neural network model is implemented based on Keras, using TensorFlow as the Keras backend, and some algorithms are implemented based on scikit-learn.

The data set used in the experiment comes from the DSD100 and MedleyDB data sets, and 50 songs are selected, covering a variety of music styles such as Rap, Country, Hip-Hop, and Rock. Each song uses music sung by 1–3 professional singers. For the humming audio file of each song, first, detect the starting points of the notes to obtain multiple starting points of notes, then randomly select the starting points of the notes to start, cut out 180 10-second segments, and finally get a total of 9,000 humming records. It is divided into a training set, validation set, and test set with a ratio of 4:1:1. Since this part of the test, the set is sung by professional singers, it is called the professional group 40 test set. In addition, for the 10 songs, 30 audio clips hummed by three students were recorded by themselves to form the non-professional group test set.

Three types of evaluation indicators are used in the experiment: accuracy rate (Accuracy, ACC), the response time (TIME), and mean reciprocal ranking (Mean Reciprocal Rank (MRR)). Since the deep-learning framework for humming recognition can give multiple recognition results and their corresponding probabilities, only using the recognition accuracy rate for evaluation cannot fully reflect the performance of the framework. For this reason, MRR is used as one of the evaluation indicators. MRR is currently widely used. Applied in the evaluation of problems that return multiple results, it can reflect the pros and cons of the returned result set. Its formula is as follows:

\[
    \text{MRR} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{r_{i}},
\]

where \( N \) is the number of experiment objects and \( r \) is the rank number in the \( r \)th experiment.

3. Results and Discussion

3.1. Humming Recognition Deep-Learning Framework and Design. The humming recognition deep-learning framework consists of the following parts:

1. Humming Audio Database. Including humming recognition training dataset and test dataset. The vocal track audio of the original singer (or singing professionally) of the song constitutes the training dataset. In addition to a part of the above audio, the test data set also contains a part of nonprofessional humming audio data to compare and evaluate the generalization ability of the model;

2. Preprocessing Module. The humming audio data is processed and analyzed to obtain the characteristic representation of the audio data. The main processing flow has been described in the third chapter;

3. Neural Network Training Module. Input the training data set into the humming recognition neural network, train in batches, use the validation set to calculate the loss function value after each iteration, stop training after reaching a certain accuracy requirement, and use the neural network with the smallest loss function value. The weight value is used as the optimal parameter output of the model;

4. Neural Network Test Module. Based on a certain amount of test humming audio data, appropriate evaluation indicators are used to test and evaluate the performance of the neural network, as the basis for repeatedly adjusting the neural network training process and parameter selection;

5. Humming Recognition System. The humming recognition system is based on a trained neural network model and accepts human humming as input. The humming signal undergoes audio processing steps such as digitization, filtering, pre-emphasis, windowing, calculation of Mel cepstral coefficients to obtain the mel-spectrogram representation of the humming signal, input the neural network...
(6) Using modular design, the deep-learning framework for humming recognition has good scalability, and several modules with high cohesion and low coupling are obtained. For example, the audio database can be flexibly replaced or modified for neural network model training, so that training evaluation can be performed on different datasets; it is also possible to use a set of model parameters trained by the neural network training module to perform training on different datasets. Testing of network performance and end-to-end testing of prototype systems are shown in Figure 2.

The humming recognition neural network model is the core part of the deep-learning framework, and its overall design ideas are as follows:

(1) The input layer receives the mel-spectrogram of the humming audio signal as input
(2) Then, uses several convolutional layers to learn the local features of the audio signal to obtain the audio signal feature map
(3) Then, several recurrent layers are used to induce and learn the sequence features of the audio signal over time
(4) Finally, the probability distribution of the recognition result that the input audio signal is a certain song is obtained through the Softmax activation function

The humming recognition deep neural network model designed accordingly is shown in Figure 3.

The data received by the input layer is a two-dimensional mel-spectrogram representation of the audio signal. In the hidden layer part, in order to fully extract the spectral features of the audio signal, four layers of convolution and pooling layers are used to learn the local features of the signal, and a threshold control unit is used in the last two layers, learning, and induction of sequence features of audio signals. Finally, the Softmax activation function is used in the output layer to output the neural network calculation result as the probability distribution of song recognition.
3.2. A Prototype System for Recognition of Humming Audio Scores. The humming audio music score recognition prototype system adopts the C/S architecture, the server is based on the pythonweb framework Bottle, the client is implemented based on the ReactNative framework, and the client and the server communicate based on the HTTP protocol. The overall interaction process design of the system is shown in Figure 4.

The user inputs the humming signal through the client recording module, and the client records the humming audio with a sampling frequency of 16000 Hz, performs a series of preprocessing, and sends an HTTPS POST request through the audio uploading module to upload the audio to the server for processing. The server-side audio preprocessing module performs note start point detection and audio segmentation on the humming audio and then inputs the humming recognition module. This module uses the humming recognition, neural network model, to give the recognition result and returns the corresponding score image for client display.

Considering that the server needs to compile the Keras model, read the trained neural network parameters and recognize the humming audio, the server is written in Python. The bottle is a lightweight Python Web framework that provides basic routing, encapsulation of request objects, template support, etc. It can realize the rapid development of small Web applications and meet the needs of the humming recognition prototype system server. The ReactNative framework adopted by the mobile terminal is a cross-platform mobile application development framework launched by Facebook, which can quickly develop mobile applications based on the React ecosystem.

3.3. Training the Neural Network. In the neural network training process, the humming audio is sampled with a sampling frequency of 16000 Hz. For each training process, the network weights are initialized with a uniform distribution, and the cross-entropy function is used as the loss function, and then the stochastic gradient descent algorithm is used for learning.

For the overfitting problem, the method of early termination of training is used in the experiment. In the training process, generally speaking, as the number of training increases, the error on the training set will be smaller, while the error on the validation set will first decrease and then increase, that is, the model begins to enter the overfitting state, and before that, there will be a good number of training cycles. Therefore, if the error accuracy set on the validation set is not improved in 10 generations of training, it means that the best number of training iterations has been reached, the training process is terminated in advance, and the network weights of the generation training with the best accuracy are reserved as a result of training. In addition, the Dropout method is also used to alleviate the overfitting phenomenon. Dropout is an extremely effective and simple regularization technique that can significantly reduce network parameter overfitting during training. Dropout can be visually understood as performing sub-sampling operations in a fully connected neural network and only updating the weights and parameters of the sampling network during training.

During model training, Dropout randomly selects some nodes in the network and sets them in an inactive state. The inactive nodes will not participate in the operation during
this training process, and their weights will not change. In the next training, repeating this process, nodes that were not activated last time may resume work. In the training process, the output of some neurons is randomly set to 0, making it invalid, so that each neuron does not completely depend on other neurons so that more feature expressions can be obtained.

During the experiment, the Dropout ratio was selected as \( p = 0.3 \). Table 1 shows the accurate rate variation of the model on the training set and the validation set before and after applying the Dropout method when the hyperparameters remain unchanged during ten training sessions.

It can be obtained from Table 1 that after applying the Dropout method, although the accuracy rate on the training data set has decreased, the accuracy rate on the verification data set has increased. It can be considered that adding Dropout has improved the generalization of the model to a certain extent and capabilities and mitigates overfitting.

In the humming recognition deep neural network model, there are several hyperparameters that will affect the model quality and training time, including the size of the convolution kernel, the size of the pooling kernel, the number of training cycles (epoch), the batch size (batch size), learning rate, momentum factor, and the dropout ratio described above.

Among them, the size of the convolution kernel and the size of the pooling kernel are the network structure parameters. A training cycle refers to the process of training all sample data once. If the number of training cycles is too small, the model may not converge, and if it is too large, overfitting may occur. The batch size is the number of data blocks selected in each stochastic gradient descent. Usually, for relatively small datasets, the entire dataset can be input into the network for training, and the resulting gradient descent direction can better represent the overall characteristics of the dataset.

For the humming recognition dataset, due to the large memory usage of audio data, it is not feasible to load all the data at one time. It is very important to use a reasonable batch size to make the sample distribution reasonable in each iteration. At the same time, within a reasonable range, increasing the batch size can reduce the number of iterations of model training and speed up the training speed of the model. The learning rate is the weight of the negative gradient in the stochastic gradient descent algorithm. Using a larger learning rate can speed up the training of the network, but it is easy to miss the minimum value and cause the model to fail to converge. When using a smaller learning rate, the network becomes slow to train and may get stuck in local minima. The momentum factor is a hyperparameter used in stochastic gradient descent to control the influence of the previous weight update on this weight update, and it also has a great impact on the training speed of the network.

In the training of the humming recognition neural network model, grid search is used to assist in obtaining good values for the hyperparameters. The principle of grid search is relatively simple. First, for each hyperparameter, set a small optional set, for example, set the optional set \([0.01, 0.02, 0.05, 0.1]\) for the learning rate. Then, grid search will perform the Cartesian product of these hyperparameter sets to obtain multiple sets of hyperparameters, and automatically use each set of hyperparameters to conduct training experiments on the model, and obtain the hyperparameter combination with the smallest validation set error as the most excellent hyperparameter selection. The final selected model hyperparameter values are shown in Table 2.

### Table 1: Suppress overfitting using the Dropout method.

<table>
<thead>
<tr>
<th></th>
<th>Mean value of ACC for 10 times</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Before dropout</td>
</tr>
<tr>
<td>Training dataset</td>
<td>0.9127</td>
</tr>
<tr>
<td>Test dataset</td>
<td>0.9016</td>
</tr>
</tbody>
</table>

### Table 2: Humming recognition neural network hyperparameters.

<table>
<thead>
<tr>
<th>Hyperparameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convolution kernel size</td>
<td>(4,4)</td>
</tr>
<tr>
<td>Pooling kernel size</td>
<td>(2,2)</td>
</tr>
<tr>
<td>Number of training cycles</td>
<td>60</td>
</tr>
<tr>
<td>Batch size</td>
<td>190</td>
</tr>
<tr>
<td>Learning rate</td>
<td>0.03</td>
</tr>
<tr>
<td>Momentum factor</td>
<td>0.9</td>
</tr>
<tr>
<td>Dropout rate</td>
<td>0.4</td>
</tr>
</tbody>
</table>

3.4. Performance Analysis of Humming Recognition Neural Network. The humming recognition neural network model with better test results is obtained by training. The composition of each network layer, the output dimension, and the number of trainable parameters are shown in Table 3.

The accuracy and loss function value (LOSS) of the network model on the training set and validation set are shown in Table 4.

On both the training set and the validation set, the model achieved a high recognition accuracy rate of more than 93%, and at the same time, the loss function value was relatively small. It can be seen that the model has fully fitted the training data and achieved good results on the validation data set. Next, the model is used to identify the two sets of test sets, and the experimental results are shown in Table 5.

On the professional group test set, the humming recognition neural network model achieved excellent recognition results, with a recognition accuracy rate of 0.9396 and an average reciprocal ranking of 0.9633. On the nonprofessional group test set, the accuracy rate has dropped to 0.7896, but it still achieves good recognition results. In terms of recognition efficiency, the average processing time of each fragment is about 0.6 s. On the whole, the proposed convolutional recurrent neural network model can better accomplish the task of humming recognition.

The reason for the drop in accuracy on the nonprofessional group’s test set is mainly because there may be some inaccuracies in the group’s humming. In addition, judging from the fact that the accuracy of the training data set is significantly higher than that of the validation data set and the professional group test data set, the model may still...
have a certain overfitting problem, which also affects the recognition accuracy on the test set.

In terms of recognition accuracy, the deep-learning framework for humming recognition has a certain improvement compared with the most humming recognition research work. In terms of response time, the deep-learning-based humming recognition method proposed in this paper has obvious advantages over the matching-based method. This is because for the deep-learning model, the recognition process just uses the trained model parameters to perform a series of matrix operations, and the results can be obtained quickly when the GPU is used for acceleration. Compared with the matching algorithm, the calculation speed has a natural advantage.

Based on the above experimental results, the humming recognition method based on deep learning proposed in this paper can better complete the humming recognition task and has certain practicability. Compared with the traditional matching-based humming recognition method, there are certain improvements and improvements.

### 4. Conclusion

Focusing on the problem of automatic recognition of humming audio signals, this paper applies deep-learning methods to humming recognition and combines traditional audio signal processing methods to design a deep-learning framework for humming audio recognition. The feasibility of the humming recognition model finally realized a deep-learning-based humming audio score recognition system. The main conclusions are as follows.

1. Aiming at the problem of using humming audio signals in a deep-learning framework, this paper studies several techniques in the field of audio analysis, including audio sampling, filtering, pre-emphasis, two-dimensional representation of signals, and note onset detection. The differences and advantages and disadvantages of the two methods provide theoretical basis and processing methods for converting humming audio signals into deep neural network input vectors.

2. On the basis of deep-learning principles and theoretical research, for the recognition of humming audio signals, a convolutional recurrent neural network model is designed and implemented, which fully combines the advantages of convolutional neural networks for local feature extraction and the advantages of recurrent neural networks in the advantages in sequence data processing, through the use of reasonable neural network components, lead to a deep-learning framework for humming recognition.

3. Using open-source deep-learning platforms and tools, experiments and demonstrations are carried...
out on the proposed deep-learning model. By training and testing on the test data set, and adjusting the model repeatedly, the model parameters with better effect are obtained. And through the evaluation test on the test data set, the feasibility and effectiveness of the proposed neural network model are verified, and the performance of the model is analyzed and evaluated.

(4) Based on the proposed deep-learning framework, using the server-side and mobile-side development technologies, a prototype system for humming audio score recognition was designed and implemented, including the server-side audio recognition service, mobile-side audio recording, audio uploading, and other functional modules.

In our study, limited by the limited research time, and the lack of understanding and practical experience of deep-learning-related theories, there are still some deficiencies in many aspects, which are worthy of more in-depth research. For example, a method for detecting the onset point of audio signal sentences is developed, instead of the onset point detection method used in this paper. Our methods refer to improved method of humming signal feature representation, an in-depth study of neural network design for humming recognition, and enhancement robustness of Deep Learning Frameworks for Humming Recognition [34].

Data Availability

The experimental data used to support the findings of this study are available from the author upon request.

Conflicts of Interest

The author declares no conflicts of interest regarding the present study.

References


