Hindawi
Scientific Programming
Volume 2022, Article ID 7905992, 10 pages
https://doi.org/10.1155/2022/7905992

Research Article

Vocal Music Recognition Based on Deep Convolution Neural Network

Zhuo He

Zhengzhou Normal University, Zhengzhou, Henan 450044, China

Correspondence should be addressed to Zhuo He; hezhuo@zznu.edu.cn

Received 24 November 2021; Revised 18 December 2021; Accepted 31 December 2021; Published 2 February 2022

Academic Editor: Baiyuan Ding

In order to achieve fast and accurate music technique recognition and enhancement for vocal music teaching, the paper proposed a music recognition method based on a combination of migration learning and CNN (convolutional neural network). Firstly, the most standard timbre vocal music is preprocessed by panning, flipping, rotating, and scaling and then manually classified by vocal technique features such as breathing method, articulation method, pronunciation method, and pitch region training. Then, based on the migration learning method, the weight parameters obtained from the convolutional model trained on the sound dataset CNN are migrated to the sound recognition, and the convolutional and pooling layers of the convolutional model are used as feature extraction layers, while the top layer is redesigned as a global average pooling layer and a Softmax output layer, and some of the convolutional layers are frozen during training. The experimental results show that the average test accuracy of the model is 86%, the training time is about 1/2 of the original model, and the model size is only 74.2 M. The F_1 values of the model are 0.88, 0.80, 0.83, and 0.85 in four aspects, such as breathing method, exhaling method, articulation method, and phonetic region training, etc. The experimental results show that the method is efficient for voice and vocal music teaching recognition. The experimental results show that the method is efficient, effective, and transferable for voice and vocal music teaching research.

1. Introduction

The concept of vocal music, also known as artistic singing, is a musical performance art that uses the combination of artistic language (singing dream) and scientific singing voice (artistic voice) to create a vivid and pleasant auditory image, singing voice, to express the highly condensed lyrics (poems or words) and typical, emotional melodic tones (good learning song) to learn vocal music, to express thoughts and feelings, and to create a second degree [1–3]. In short, vocal music is music with language sung and the human voice. Vocal music includes American singing, Gregorian chant, folk singing, and popular singing, as shown in Figure 1.

With the continuous development of economy, people's needs for material and spiritual aspects become more comprehensive and high level. With the continuous reform and development of education and the integration of various arts into people's daily life, people's appreciation level of vocal art has gradually increased, which has put forward higher requirements for the vocal art itself. High-quality vocal music appreciation and tasting are given better requirements. In order to improve the technical level of vocal teaching, this requires better technical development skills and error correction. The improvement of vocal technique is directly related to the content of vocal art. Therefore, the exploration of the status and role of vocal technique in vocal art can help the further development of vocal art by providing a better understanding of the current situation and future direction of vocal art. However, there are still many problems in vocal music teaching, such as low learning efficiency and ineffectiveness. Therefore, a new method needs to be found.

Convolutional neural network method is considered as a good learning and training method [4–10]. Deep learning theory was first proposed by Hinton et al. [11–14] as an effective method to simulate the sound learning process of brain recognition; it shows big advantages in vocal music processing and pattern recognition. For example, Janssens

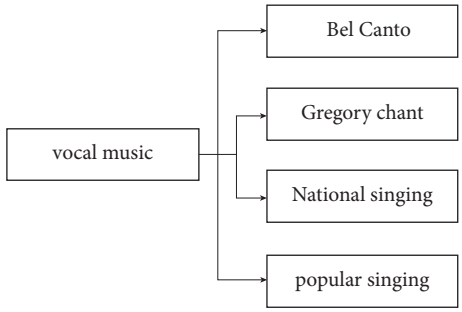

Figure 1: Vocal classification.

et al. [15, 16] proposed an end-to-end convolutional neural network based on the frequency domain representation of vibration signals to achieve fault classification of bearings with an accuracy of 93.6%. 97.58% accuracy was achieved by Qiao et al. using a convolutional neural network model to process SAR type images for classification. Qin et al. [17] gave a deep dual convolutional neural network in order to extract multidomain information of vibration signal and achieved 97.02% accuracy for engine misfire diagnosis. Zhang Kang et al. [18] proposed a deep convolutional neural network with random discard and batch normalization to accurately identify engine misfire faults based on the original cylinder head diagnostic signal. From the image processing and pattern recognition mentioned above, it can be seen that CNN models have powerful automatic feature extraction capability to perform deep feature extraction on signals [19–26] with stronger robustness and better generalization capability. Meantime, a number of parameters are greatly reduced by weight sharing and pooling operations, which reduces the training cost. In terms of pattern recognition of voice, most scholars use vibration signal for pattern recognition of voice, which has an impact on the online diagnosis of educated voice due to the high sampling rate of vibration signal and thus the relatively large scale of the constructed network. Considering that the frequency signal is easy to collect and stable, and the sampling rate is relatively low, the constructed network is easier to realize the online diagnosis. Therefore, this paper proposes the method of using CNN to automatically obtain the vocal features of the sound signal and then compare it with the standard music, write the algorithm into the STM32 microcontroller, and analyze it with the standard vocal music recorded in it to quickly give the vocal defects of the singer.

2. CNN Contrast Diagnosis Algorithm

A CNN is a feedforward neural network with convolutional operations. Compared to fully CNN of the same size, CNN has local connectivity, weight sharing, and downsampling [27] and therefore requires fewer parameters and memory for network training. The composition of neural network includes input layer, hidden layer, and output layer. The loss of forward propagation is transmitted forward by the input layer, and then the activation function is defined. On this basis, the class of the two-layer neural network is defined, the activation function is initialized, and the weights are used to realize the forward propagation of the core training logic of the neural network and calculate the loss and mutual propagation update. The final prediction process is a process of calculating the output value forward. The above is the core process of the whole calculation.

2.1. Two-Dimensional Convolution. Convolution is one of the core mathematical operations in CNN, mainly used to extracts more abstract feature from the data. Convolution layers convolve the local area of the input signal with the convolution kernel and then add the corresponding bias to the convolution output and perform a nonlinear transformation by the activation function to get the corresponding feature map. For two-dimensional linear non-shift systems, the output sequence $y(m, n)$ is equal to the convolution sum of the input sequence, $X(m, n)$, and the unit impulse response sequence $H(m, w)$. The one-dimensional wave signal is decomposed by wavelet packet to obtain multiple groups of wavelet coefficients, and then these wavelet coefficients are arranged into a two-dimensional matrix as the input of depth learning algorithm. The main calculation procedure is shown as follows:

$$a^l = f\left(\sum_{iem} x_i^{l-1} \otimes w_i^l + b_i^l\right), \tag{1}$$

where x_i^{l-1} is the element in the convolution region of the i ck (convolution kernel) of the l-1st layers; w_i^l is a weight of the i ck of the l layers; $\otimes$ is the convolution operation; b_i^l is the bias of the i-th ck of the l-th layers; M is the convolution region; and a^l is the output of the l-th layers convolution after the action of the activation function $f(-)$.

In order to avoid accelerate the learning speed of the network, the activation function adopted in this paper is Rectified Linear Unit (ReLU (x)), whose formula is shown as follows:

$$\text{relu}(x) = \max(0, x)\begin{Bmatrix} 0, & x \le 0 \\ x, & x > 0 \end{Bmatrix}. \tag{2}$$

x indicates the input value.

2.2. Maximum Pooling. The main purpose of max pooling is to compress the output features of the convolutional layer and extract the main features, thus reducing the sizes of the input vector and the parameters of the convolutional NN, reducing the training time and controlling overfitting. The maximum value of the upper left region is 9, the maximum element value of the upper right region is 2, the maximum value of the lower left region is 6, and the maximum value of the lower right region is 3. In order to calculate the values of the four elements on the right, we need to calculate the 2 of the input matrix × 2. Do the maximum operation in the area. It is like applying a filter with a scale of 2, because we chose 2 × Zone 2, stride 2; these are the hyperparameters for maximum pooling. Because the filter we use is 2 × 2, the final output is 9. Then move 2 steps to the right to calculate the

maximum value of 2. Then, in the second line, move down 2 steps to get the maximum value of 6. Finally, move 3 steps to the right to get the maximum value of 3. This is a 2×2 matrix, i.e., $f = 2$; step length is 2, i.e., $s = 2$. This is an intuitive understanding of the maximum pooling function. You can put this 4×4 region as regarded as the set of some features, that is, the set of inactive values of a certain layer in neural network. In this paper, we adopt the maximum pooling method as shown in Figure 2. K-maxpooling means that the original max pooling over time only takes the strongest value from a series of eigenvalues of the revolution layer, so our idea can be expanded. K-MAX pooling can take the value scored in the top-K of all eigenvalues and retain the original order of these eigenvalues (Figure 3 is the schematic diagram of 2-max pooling). That is, more feature information is reserved for subsequent stages. You can implement a minpooling layer and embed it into any network to see how the effect is. Computer science is based on practice. Why is there no minpooling? Suppose that the minpooling layer is used to replace the maxpooling layer. Generally, the maxpooling layer is an active layer before it, such as the ReLU layer, which will make the value of convolution characteristics greater than or equal to 0. If you use minpooling again, the resulting activation graph is the minimum value in the neighborhood of 0 or close to 0. When you go through minpooling several times, you will find that all activation values are 0, so your network cannot train, because any useful information is gone. Maxpooling is now commonly used to reduce the dimension of features and retain the maximum response of low-level features such as edges and textures after the first convolution layer and convolution block of convolution network. The pooling process can be expressed as shown in Figure 2.

2.3. Fully Connected Layer after Convolution and Pooling. Take vocal music for example. The size of a piece of sound is width × height × channel number (generally three-color channel). Assuming that the size of a group of sounds is n, if we use the traditional neural network to deal with this vocal music, the input layer needs n neurons, and if the fully connected structure is adopted, there will be many weight parameters, which is very difficult and time-consuming for the training of the network. Moreover, if the image is very complex, it is impossible to capture more advanced sound features by increasing the number of hidden layers, because neural networks with too many hidden layers may have problems in gradient back propagation, such as gradient explosion and gradient disappearance. Therefore, the traditional neural network is not suitable for dealing with vocal music tasks. In contrast, the convolution neural network adopts the design idea of local connection, weight sharing (that is, the convolution kernel is only connected with one window, and the convolution kernel can be shared by multiple windows), and pooling. The superposition of the three strategies greatly reduces many, many unnecessary weight parameters in the network, making network training easier. It should be pointed out that CNN using gap instead of FC usually has better prediction performance. The final classification of the network is achieved at network by the fully connected layer. The results of the fully connected layer are calculated as

$$z^l = f\left(w^l x^{l-1} + b^l\right). \tag{3}$$

In equation (3), w^l is a weight of the lth layers; x^{l-1} is the output value of the lth layers; b^l is the bias of the lth layers; $f(-)$ is a activation function, and a ReLU activation function is used; and z^l is the output value of the lth layers. The whole connection layers process can be expressed as follows.

2.4. Classification Evaluation and Loss Function for the Classification of Vocal Patterns. Scoring function is widely used in structure based computational aided drug design. It provides a theoretical basis for drug efficacy evaluation in drug research and development by quantitatively evaluating drug target interaction [1–5] and improves the efficiency of screening active compounds. Quantitative evaluation of the interaction between drugs and target proteins is usually divided into two steps. One step is docking process, which mainly refers to conformation search to find out potential binding poses; the other step is scoring process, which usually refers to scoring to predict drug target binding force. Most scoring functions are not approximated based on the complete physical model, so they often do not strictly follow the multi-body expansion theory, conservation law, symmetry invariance, etc. Even the expressions of some knowledge-based scoring functions do not contain physical meaning at all. In fact, as a tool applied in the scenario of high-throughput drug screening, most scoring functions focus on efficiency and pursue the balance between accuracy and efficiency by approximate means.

When using machine learning model to solve problems, there are two important concepts when it comes to model construction and model evaluation.

Loss function: most machine learning algorithms need to maximize or minimize a function, namely, "objective function." Generally, the minimization function is called "loss function."

Loss function is used in model construction (some simple model construction does not need loss function, such as KNN), so it is used to guide model generation. The damage function can be classified as follows (Figure 4).

Evaluation index: evaluate the machine learning algorithm model. In some problems, the loss function can be directly used as the evaluation index (for example, in regression problems, the mean square error (MSE) can be used not only to guide the model construction, but also to evaluate the model performance after the model is completed).

The evaluation index is used after the model is built, so it is used to evaluate the performance of the model. Common evaluation indicators of classification types can be expressed as follows.

Classification type common evaluation indicators: confusion matrix, accuracy, precision, recall, ROC-AUC, P-R curve. The Softmax evaluation function is used as the probabilistic output of the final classification layers of the CNN.

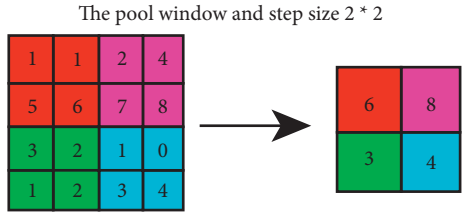

Figure 2: Schematic diagram of the largest set.

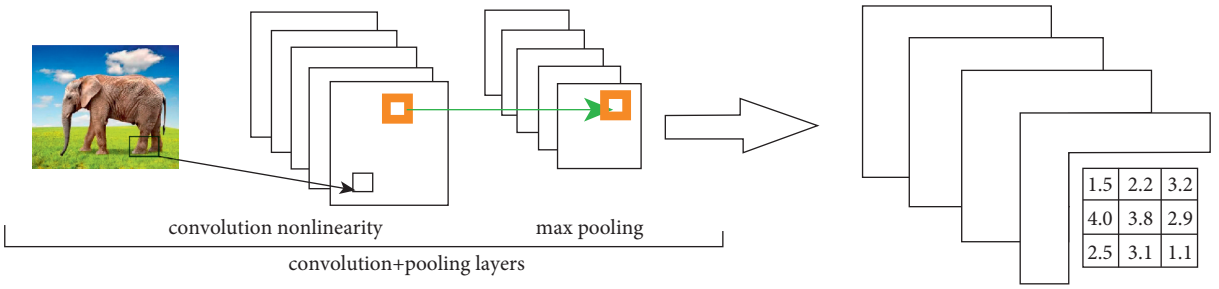

Figure 3: Process diagram of the whole connection layer.

$$p(z_j) = \text{softmax}(z_j) = \frac{e^{z_j}}{\sum_{k=1}^{N} e^{z_k}}. \tag{4}$$

In equation (4), z_j is the activation value of the jth neuron in the output layer; N is the number of vocal skill classifications; and $p(z_j)$ is the probabilistic output of each neuron.

The loss function used in this paper is the classification cross-entropy function, which is obtained by calculating the cross-entropy in the output vector after Softmax and the actual labels of the samples. The loss function is calculated as shown in the following:

$$\text{Loss} = -\sum_{i=1}^{N} y_i' \log(y_i). \tag{5}$$

In equation (5), y_i' is the i-th value of the actual labels; y_i is the i-th value of the output layers; and N is a number of vocal skill classifications. In the backpropagation stage, Adam algorithm is chosen in this paper to effectively update the weights and bias values of the network, and Adam algorithm uses first-order moment estimation and second-order moment estimation to dynamically adjust the learning rate of updating each parameter, so as to update the some weights to find the optimal solution [4, 28–30]. The standard deviation σ in data evaluation is shown as follows:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} (x_i - x)^2}{n}}. \tag{6}$$

x is the standard value and x_i is the data sample.

The more common loss functions currently in use are shown as follows:

(1) 0-1 loss:

$$L(Y, f(X)) = \begin{cases} 1, & Y \neq f(X), \\ 0, & Y = f(X). \end{cases} \tag{7}$$

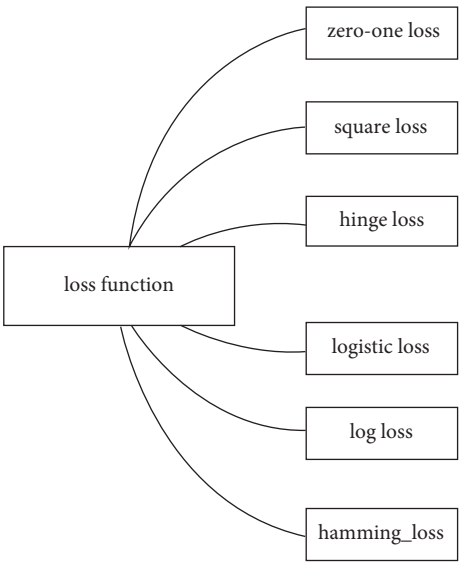

Figure 4: Classification of loss function.

0-1 loss means that the predicted value and the target value are not equal to 1; otherwise they are 0. 0-1 loss function directly corresponds to the number of classification errors, but it is a non-convex function, which is not very applicable.

(2) Squared loss:

$$L(Y, f(X)) = (Y - f(X))^2. \tag{8}$$

This represents frequent application and regression problems.

(3) Absolute loss:

$$L(Y, f(X)) = |Y - f(X)|. \tag{9}$$

It is very sensitive to outliers and noise. It is often used in AdaBoost algorithm.

(4) Logarithmic loss:

$$L(Y, P(Y|X)) = -\log P(Y|X). \tag{10}$$

(a) Log loss function can characterize the probability distribution very well, in many scenarios especially multi-categorization; if you need to know the confidence that the result belongs to each category, then it is very suitable.
(b) Not very robust, more sensitive to noise than hinge loss.
(c) The loss function of logistic regression is the log-log loss function.

(5) Hinge loss:

$$L(Y, f(X)) = \max(0, 1 - f(X))^2. \tag{11}$$

(a) Hinge loss function means that if it is correctly classified, the loss is 0; otherwise the loss is [Formula]. SVM uses this loss function.
(b) The general [Formula] is the predicted value. Between −1 and 1, [Formula] is the target value (−1 or 1). It means that the value of [Formula] is between −1 and +1. It does not encourage [Formula]; that is, it does not encourage the classifier to be overconfident. There will be no reward for making a correctly classified sample more than 1 from the division line, so that the classifier can focus more on the overall error.
(c) It is relatively robust and insensitive to outliers and noise, but it does not have a good probability interpretation.

(6) Loss function of LR:

$$L(Y, \pi(X)) = -Y \log \pi(X) - (1 - Y)\log(1 - \pi(X)). \tag{12}$$

2.5. Network Structure of the Convolutional Neural Network. CNN is a variant of multilayer perceptron (MLP). It was developed by biologists Huber and Wiesel's early research on cat visual cortex. The cells in the visual cortex have a complex structure. These cells are very sensitive to subareas of visual input space, which we call receptive fields, and tile the whole field of vision in this way. These cells can be divided into two basic types, simple cells and complex cells. Simple cells respond to the marginal stimulation pattern in the receptive field to the greatest extent. Complex cells have larger acceptance domains, which is locally invariant to stimuli from the exact location, from NN to convolutional neural network, as shown in Figure 5.

This tight relationship between interlayer connections and null domain information in CNNs makes them suitable for vocal processing and understanding. Moreover, they have also shown superior performance in automatically extracting salient features of the voice. In an example, GF (Gabor filters) have been used in initialization preprocessing step to achieve simulating the response of the visual system to visual stimuli. In most of the current work, researchers have applied CNNs to a variety of machine learning problems, including sound recognition, document analysis, and language detection. For the purpose of finding frame-to-frame coherence in sound, CNNs are currently trained by a temporal coherence, but this is not specific to CNNs.

In fact, according to the training results in the document, you may get a lot of such conclusions. For example, the neuron on the first floor is the simplest classifier. What it does is to detail whether there are green, yellow, and oblique stripes. The second layer is more complicated than this. According to the output of the first layer, it can see that the straight line and horizontal line are part of the window frame, the brown grain is wood grain, and the diagonal stripe + gray may be many things (part of the tire, etc.). According to the output of the second hidden layers, the third hidden layer will do more complex things. But the problem now is that when we directly use the fully connect feed forward network for image processing, we often need too many parameters. For example, suppose this is a $100 * 100$ color map (a small image), you pull this into a vector (how many pixels does it have), and it has $100 * 100\,3$ pixels. If it is a color graph, each pixel needs three values to describe it, that is, 30,000 dimensions. If the input vector is 30,000 dimensions, assuming that the hidden layer has 1000 neurons, the parameters of the hidden layer are $30{,}000 * 1000$, which is too much. So what CNN does is simplify the architecture of the neural network.

We know from human knowledge and from our images that some weights are not useful, and we filter them out at the beginning. Instead of using fully connect feed forward network, it uses relatively few parameters for image processing, so CNN is simpler than ordinary DNN.

After we finish our talk, we will find that you may think the operation of CNN is very complex, but in fact, its model is simpler than DNN. We use power knowledge to remove some parameters from the original fully connect layer and become CNN. Suppose that the input of our network is a $6 * 6$ image. If it is black and white, a pixel only needs a value to describe it. 1 means that ink is applied, and 0 means that ink is not applied. In the revolution layer, it consists of a group of filters (each filter is actually equivalent to a neuron in the fully connect layer), each filter is actually a matrix $(3 * 3)$, and the parameters in each filter (each element value in the matrix) are network parameters (these parameters need to be learned and not designed by people). If each filter detects $3 * 3$, it means that it detects another $3 * 3$ pattern (see a range of $3 * 3$). When detecting a pattern, you can decide whether a pattern appears by looking at only a range of $3 * 3$ without looking at the whole image. This is the first property we consider, Stripe = 1, which can ensure that every part of the image can be convoluted to $6 - 3 + 1 = 4$. The filter will tell you that the maximum value ($3 * 3$ matrix inner product) at the top left and bottom left represents that the pattern to be detected by the filter appears in the upper left corner and lower left corner of the image. This matter is

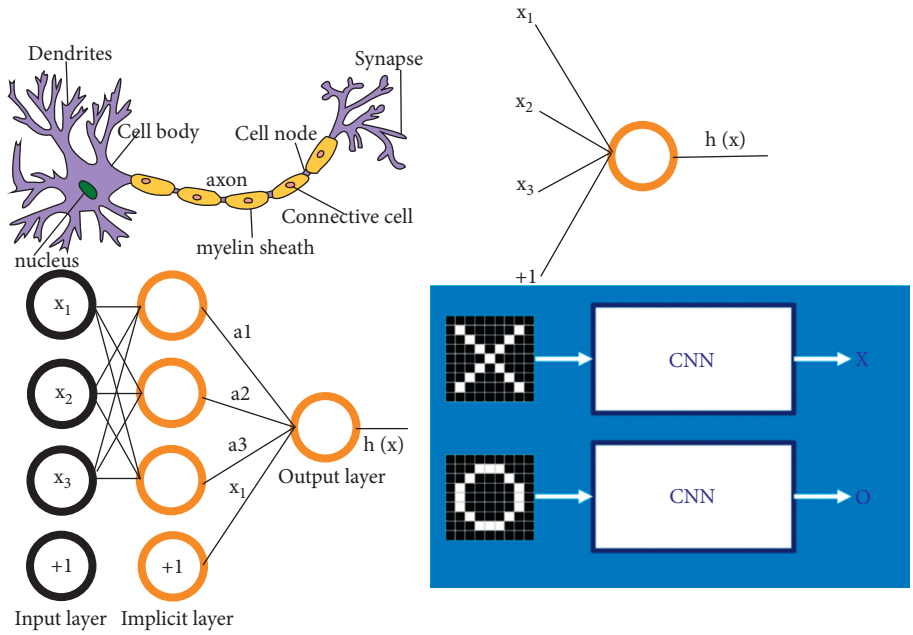

Figure 5: Neural network to convolution neural network process diagram.

considered as property 2. If the same pattern appears in the upper left corner and the lower left corner, we can use filter 1 to detect it. We do not need different filters to do this. Revolution is fully connected to remove some weight. Set the $6 * 6$ image flag as vector (36) and weight as filter (marked with different colors), such as 1, 2, 3, 7, 8, 9, 13, 14, 15 (9 each time, originally 36 each time). Different neurons share one weight (shared weight). Reducing weight and sharing weight can reduce parameter max pool down sampling pool: reduce the amount of calculation, but lose information. Here is a question: for the first time, there are 25 filters to get 25 feature maps. For the second, there are also 25 filters. Do you want to get a $25 * 25$ feature map? It is not like that! Suppose there are two filters in the first layer, and the filter in the second layer will consider the depth when considering this input. Instead of considering each channel separately, all channels are considered at once. Therefore, the output has as many filters as the revolution has (the revolution has 25 filters and the output has 25 filters; however, these 25 filters are a cube).

This greatly reduces the parameter scale of NN architecture. The designed CNN has 8 layers, including input layers, two sets of alternately connected convolutional and pooling layers, tiling layer, fully connected layers, and output layers, as shown in Figure 6.

The specific signal processing is shown in Figure 7.

The features are extracted through the convolution layer, and then the convolution results are mapped nonlinearly using the ReLU activations function. The pooling layers are then processed to eliminate the redundancy of information and reduce the number of model parameters. Finally, feature classification is performed by a fully connected layer and a Softmax output layer. The loss function used in the network is the classification, and the weights of the network are updated using the Adam optimizer. The structural parameters of the CNN model after iterative optimization are shown in Table 1.

Vocal voice recognition is often carried out after time-frequency analysis to obtain the speech spectrum. Among them, the speech sequence spectrum is characterized by a sequence of waves. In order to increase the effect of voice identification, it is necessary to overcome various characteristics of voice signal, including vocal breathing method, enunciation method, pronunciation method, and sound area training. CNN provides convolution in time and space. The idea of CNN is applied to the music modeling of speech recognition. Convolution can be used to overcome the diversity of speech signals. Signal is regarded as a wave and is widely used in deep convolution network recognition in sound. Vocal music and speech recognition are similar to the specific process in Figure 8.

The main algorithm flow is as follows:

(1) Standard vocal preprocessing. The collected high quality vocal music is preprocessed by panning, flipping, rotating, scaling, etc. to realize the expansion of the dataset. And the vocal music is uniformly adjusted to different frequencies and peaks.

(2) Input vocal music samples. Three vocal music samples are randomly selected from educators' vocal music styles as training samples input.

(3) Construct vocal music education recognition model. Based on the overall architecture of the pretrained VGG16 model, the original Softmax classification layer is replaced with a Softmax classifier with 6 neurons and the remaining fully connected layers are replaced in a global average pooling layer.

(4) Parameter migration and fine-tuning. Initialize the parameters of the pretrained VGG16 model by

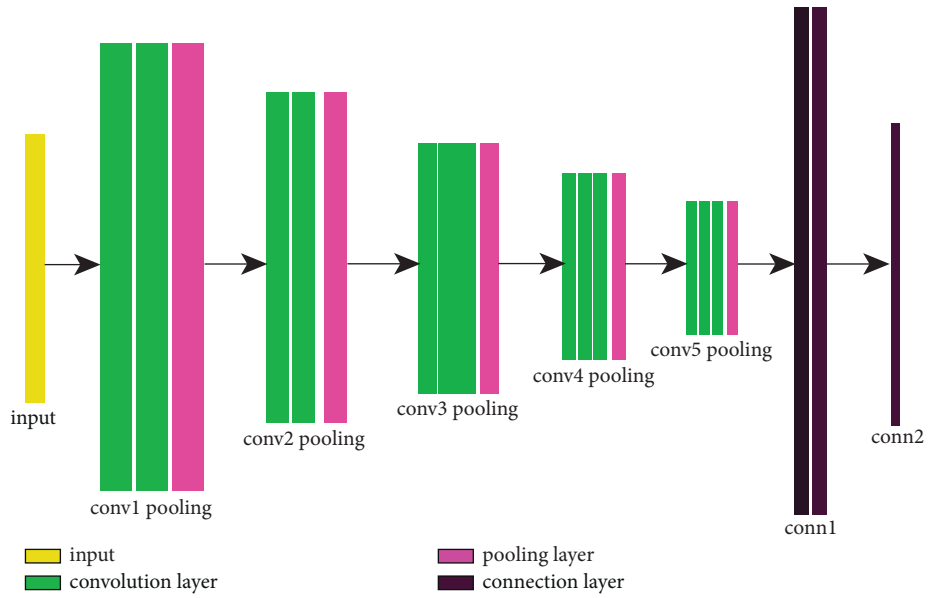

Figure 6: Convolutional neural network structure diagram.

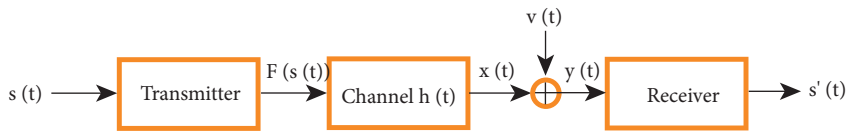

Figure 7: Spectrogram as observed signal frequency spectrum.

Table 1: Structural parameter of CNN.

Serial number	Neural network layer	Nuclear size	Number of nuclear	Step length	Activation function	Make-up zeroes	Output size
1	Input layer	—	—	—	—	—	[116, 8, 1]
2	Convolutional layer C1	6 * 2	16	1 * 1	ReLU	SAME	[116, 8, 1]
3	Pooling layer S1	3 * 2	16	3 * 2	—	SAME	[39, 4, 16]
4	Convolutional layer C2	6 * 2	32	1 * 1	ReLU	SAME	[39, 4, 36]
5	Pooling layer S2	2 * 2	32	2 * 2	—	SAME	[20, 2, 32]
6	Flat layer	—	—	—	—	—	1280
7	Fully connected layer	1	1	—	ReLU	—	60
8	Output layer	1	1	—	Sigmoid	—	7

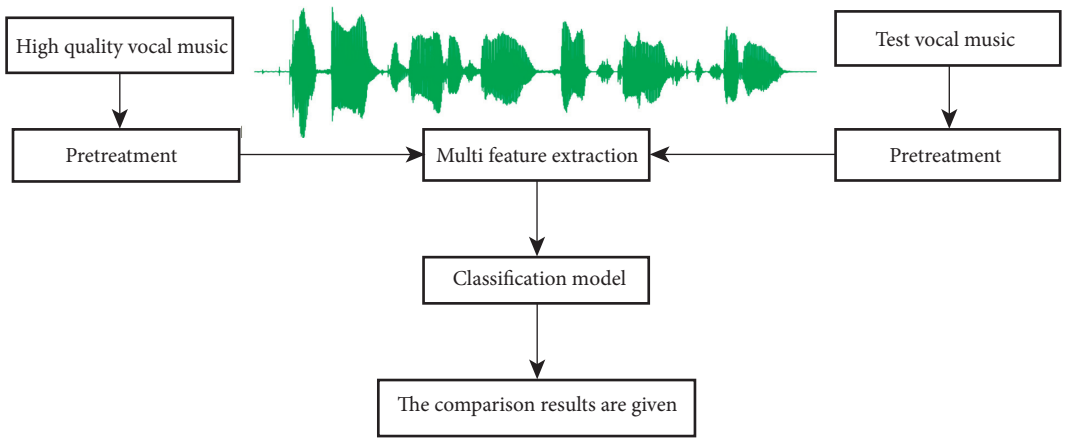

Figure 8: Convolutional neural network audio signal processing technique.

parameter migration, set the training parameters such as learning rate, momentum factor, and batch size, freeze the parameters of the pooling layer and some convolutional layers by updating the loss function iteratively, and update and optimize the parameters of some convolutional layers and Softmax classification layers.

(5) Testing of the model. The model is tested by taking performance pieces from the educated human dataset as test samples to verify the accuracies of the model.

3. Results and Analysis

There are two ways of migration learning: freezing all the convolutional layers and fine-tuning some of the network layers. Freezing all the convolutional layers means that the top layer of the model is allowed to participate in the training and all the convolutional layers are not involved. Freezing all the convolutional layers means that the top layer of the model is allowed to participate in the training, and none of the convolutional layers are involved. In this experiment, three scenarios are designed to analyze the effectiveness of migration learning by freezing all the convolutional layers, depending on the structure of the custom top layer: (1) dense layers with 64 neurons and Softmax output layers; (2) dense layers and Softmax output layers; (3) using the original top layer structure of VGG16; (4) pooling layers and Softmax output layers. The experimental results obtained for the three samples are shown in Table 2.

The relationship between the variation of the vocal value and the effective value at each time point during the experiment is shown in Figure 9.

It is obvious to see that different vocal test results have certain errors from the valid values, and different test samples can get the errors and losses from the valid values, so that the effect of each vocal sample can be analyzed and the problems in vocal music can be fed back, mainly through the four characteristics of vocal music (breathing method, articulation method, pronunciation method, and pitch range training) methods to improve. The graph below shows the errors values between the different characteristics and the valid standard samples, as shown in Figure 10.

The purpose of our vocal music teaching research is to quickly improve the vocal music level of the trainees. The difference between the trainees and the standard value or standard vocal music shown in Figure 10 will help to quickly make up for or improve the shortcomings in vocal music teaching. This shows that there are still great differences in our trainees. In ordinary education and teaching, we should further understand and train for this difference.

The stagger analysis of the training data is shown in Figure 11:

Through the errors in the training, we can determine the key points and special projects in the real vocal music training, which will help us make different training plans for each trainer and ensure the reasonable learning of vocal music technology, as shown in Figure 12.

Through the error change, we can find that the difference between the sample value and the standard value is changing

Table 2: Testing and accuracy.

Experiment number	Accuracy (%)	Test time (s)
1	78	30
2	90	30
3	86	30

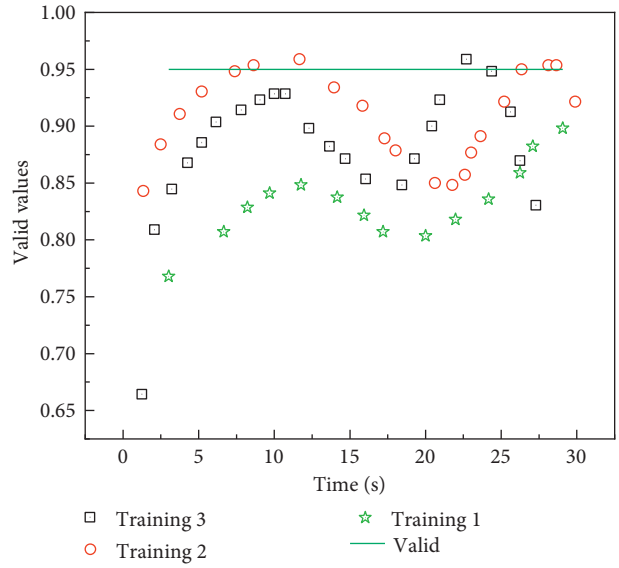

Figure 9: Standard errors for different test samples.

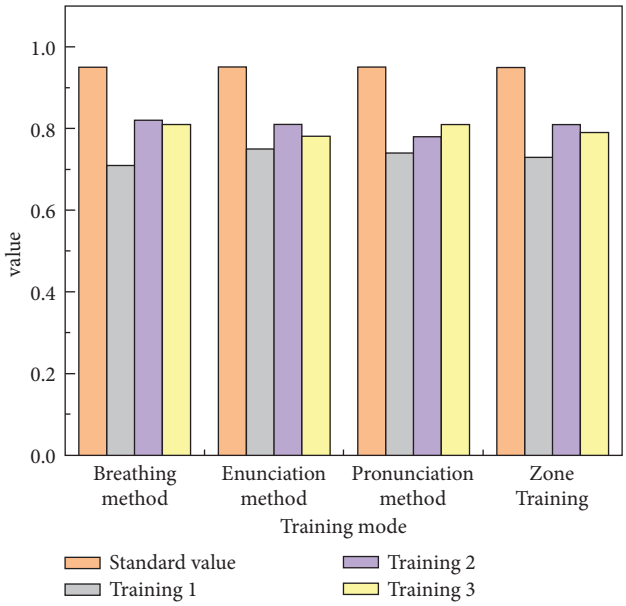

Figure 10: Error situation under different characteristics.

in the whole process, which is what we need to improve in the error learning. Only through the continuous error change to find the error can we continuously improve the ability and skills of vocal music learners in vocal music teaching, so as to fundamentally improve the level of vocal music teaching.

The comparison between the test sample and the standard sample can be evaluated for each segment and the gaps that exist

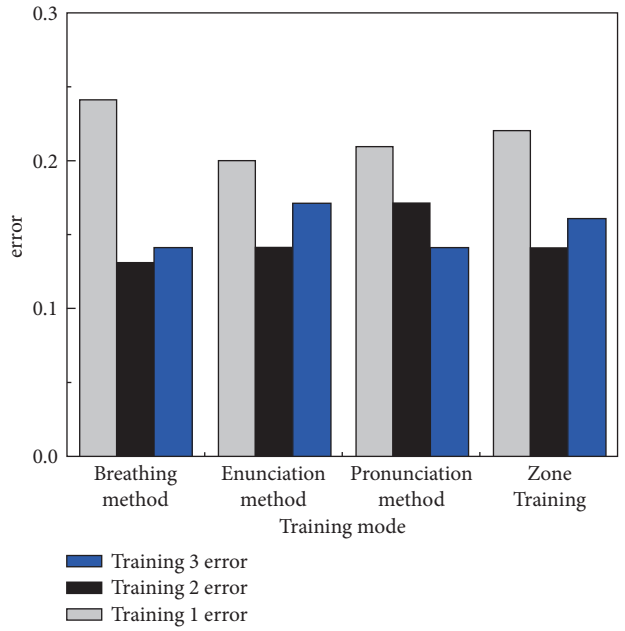

Figure 11: Error training chart.

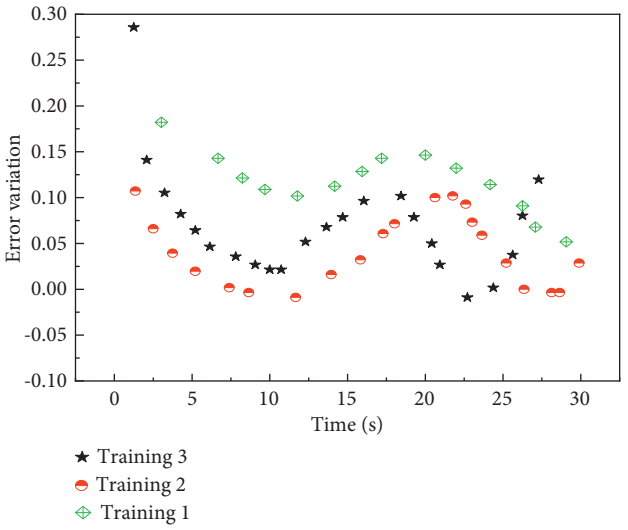

Figure 12: Error variation curve.

in the four characteristics of vocal music can be analyzed. This allows for timely targeted training tests for the gaps that exist in order to achieve rapid improvement, which is the application of the neural network method in vocal music teaching. It is also an important method for research in vocal music teaching.

In summary, this result shows that the standard sample convolutional neural network for uninvolved training samples for the control obtained the corresponding accuracy. And the accuracy of different characteristics was analyzed.

4. Conclusion

For the traditional recognition methods, there are problems such as poor recognition effect and low efficiency. In this paper, we introduced migration learning, redesigned the top layer of the model, compared the recognition effect of different sampling schemes and whether to freeze all convolutional layers during training, and obtained the following conclusions. In this study, the human voice features were analyzed and modeled by CNN method, and the accuracy and error under four features were predicted and combined with the development of the change curve of the error (0.1–0.3). The results show that the deep learning method can obtain better model results giving a quick evaluation with what is not known and giving relevant training suggestions for the trainer. It can quickly improve the vocal skills of learners and the vocal characteristics of vocal educators and helps learners to target their deficiencies with special training.

Data Availability

The dataset can be accessed upon request to the author.

Conflicts of Interest

The author declares no conflicts of interest.

References

[1] J. Sun, "Research on resource allocation of vocal music teaching system based on mobile edge computing," *Computer Communications*, vol. 160, no. 2, 2020.

[2] J. M. P. Wilbiks, D. T. Vuvan, P. Y. Girard, I. Peretz, and F. A. Russo, "Neurocase. Effects of vocal training in a musicophile with congenital amusia," *Neuron*, vol. 33, no. 2, pp. 1–191, 2020.

[3] Y. L. Huang, S. Y. Meng, X. S. Li, and W. Y. Fan, "A classification method for wood vibration signals of Chinese musical instruments based on GMM and SVM," *Traitement du Signal*, vol. 35, no. 2, pp. 137–151, 2018.

[4] M. Pei, X. Wu, Y. Guo, and H. Fujita, "Small bowel motility assessment based on fully convolutional networks and long short-term memory," *Knowledge-Based Systems*, vol. 121, no. 1, pp. 163–172, 2017.

[5] J. Kawahara, C. J. Brown, S. P. Miller et al., "BrainNetCNN: convolutional neural networks for brain networks; towards predicting neurodevelopment," *NeuroImage*, vol. 146, pp. 1038–1049, 2017.

[6] M. Ghafoorian, N. Karssemeijer, T. Heskes et al., "Location sensitive deep convolutional neural networks for segmentation of white matter hyperintensities," *Scientific Reports*, vol. 7, no. 1, 2017.

[7] M. T. Mccann, K. H. Jin, and M. Unser, "Convolutional neural networks for inverse problems in imaging: a review," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 85–95, 2017.

[8] D. S. Tan, W. Y. Chen, and H. Kai-Lung, "DeepDemosaicking: Adaptive image demosaicking via multiple deep fully convolutional networks," *IEEE Transactions on Image Processing*, vol. 25, no. 5, 2018.

[9] Z. J. Wu and S. M. Naik, "DSP applications in engine control and onboard diagnostics: enabling greener automobiles," *IEEE Signal Processing Magazine*, vol. 34, no. 2, pp. 70–81, 2017.

[10] K. Lee, S. E. Kim, J. Doh, K. Kima, and W. K. Chung, "User-friendly image-activated microfluidic cell sorting technique using an optimized, fast deep learning algorithm," *Lab on a Chip*, vol. 21, no. 9, 2021.

[11] A. M. Stoica, T. V. Chelaru, F. Stoican, and B. D. Ciubotaru, "A Kalman filtering approach for systems subject to parametric modeling uncertainties," *IFAC-PapersOnLine*, vol. 52, no. 12, pp. 400–404, 2019.

[12] C. Gu, X.-Y. Qiao, H. Li, and Y. Jin, "Misfire fault diagnosis method for diesel engine based on MEMD and dispersion entropy," *Shock and Vibration*, vol. 2021, Article ID 9213697, 14 pages, 2021.

[13] S. Sneha, S. Potala, and A. R. Mohanty, "An improved method of detecting engine misfire by sound quality metrics of radiated sound," *Proceedings of the Institution of Mechanical Engineers-Part D: Journal of Automobile Engineering*, vol. 233, no. 12, pp. 3112–3124, 2019.

[14] X. Wang, Z. Li, H. Shan, Z. Tian, Y. Ren, and W. Zhou, "FastDerainNet: a deep learning algorithm for single image deraining," *IEEE Access*, vol. 8, p. 99, 2020.

[15] O. Janssens, V. Slavkovikj, B. Vervisch et al., "Convolutional neural network based fault detection for rotating machinery," *Journal of Sound and Vibration*, vol. 377, pp. 331–345, 2016.

[16] X. Ding and Q. He, "Energy-fluctuated multiscale feature learning with deep ConvNet for intelligent spindle bearing fault diagnosis," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 8, pp. 1926–1935, 2017.

[17] C. Qin, Y. Jin, J. Tao et al., "DTCNNMI: a deep twin convolutional neural networks with multi-domain inputs for strongly noisy diesel engine misfire detection," *Measurement*, vol. 180, Article ID 109548, 2021.

[18] K. Zhang, T. Jian-feng, C. Qin, W. Li, and C. Liu, "Diesel engine misfire diagnosis with deep convolutional neural network using dropout and batch normalization," *Journal of Xi'an Jiaotong University*, vol. 53, no. 8, pp. 159–166, 2019.

[19] J. Feng, Y. Yao, S. Lu, and Y. Liu, "Domain knowledge-based deep-broad learning framework for fault diagnosis," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 4, pp. 3454–3464, 2021.

[20] P. L. Wu, X. Y. Nie, and G. Xie, "Multi-sensor signal fusion for a compound fault diagnosis method with strong generalization and noise-tolerant performance," *Measurement Science and Technology*, vol. 32, no. 3, pp. 1–16, 2021.

[21] M. Long, J. Wang, Y. Cao, J. Sun, and P. S. Yu, "Deep learning of transferable representation for scalable domain adaptation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 8, pp. 2027–2040, 2016.

[22] C Huizhu, X. L. Liu, and J. A. Qiu, "Citespace based comprehensive analysis on debris flow risk of China during recent 30 years," *Journal of Engineering Geology*, vol. 26, no. 2, pp. 286–295, 2018.

[23] Z. G. Deng, H. D. Guan, H. John, M. A. Forster, Y. Wang, and C. T. Simmons, "A vegetation-focused soil-plant atmospheric continuum model to study hydrodynamic soil-plant water relations," *Water Resources Research*, vol. 53, no. 6, 2017.

[24] R. Duan, Y. Dong, P. Zhou, L. Wang, F. U. Yunmei, and S. Zhao, "Advances in application of hyperspectral remote sensing in hydrogeology," *Hydrogeology and Engineering Geology*, vol. 44, no. 4, pp. 23–29, 2017.

[25] Y. Zhang, S. Liu, C. Dong, X. Zhang, and Y. Yuan, "Multiple cycle-in-cycle generative adversarial networks for unsupervised image super-resolution," *IEEE Transactions on Image Processing*, vol. 29, p. 99, 2019.

[26] N. H. Son and M. S. Szczuka, "Neural networks design: rough set approach to continuous data," *Lecture Notes in Computer Science*, vol. 1263, pp. 359–366, 2019.

[27] J. Li, X. M. Wang, Y. H. Zhang, W. D. Wang, and S. J. Gai, "Research on the seismic phase picking method based on the deep convolutional neural network," *Chinese Journal of Geophysics*, vol. 63, no. 4, pp. 1591–1606, 2020.

[28] S. N. Pradhan, M. Anjum, and P. Jena, "Estimation of soil moisture content by remote sensing methods: a review," *Journal of Pharmacognosy and Phytochemistry*, vol. 7, pp. 1786–1792, 2018.

[29] P. H. Ma, J. B. Peng, Q. Y. Wang, X. Zhu, Q. Dong, and D. Zhai, "Formation mechanism, deposits and motion characteristics of the typical loess landslide in South Jingyang Platform," *Journal of Engineering Geology*, vol. 26, no. 4, pp. 930–938, 2018.

[30] S. J. Mi, Q. B. Wu, and B. Sheng, "Research on estimating soil Moisture by using thermal inertia," *Geomatics and Spatial Information Technology*, vol. 42, no. 10, pp. 11–14+18, 2019.