ISSN 2616-5775 Vol. 7, Issue 2: 33-42, DOI: 10.25236/AJCIS.2024.070205

An Improved Philips Algorithm

Wenze Qiu[1,a,*], Yiqing Xia[1,b], Qiwei He[1,c], Xu Zhao[1,d]

[1]East China University of Science and Technology, Shanghai, China
[a]21012895@ecust.edu.cn, [b]20002487@ecust.edu.cn, [c]20003954@ecust.edu.cn,
[d]21012897@ecust.edu.cn
[*]Corresponding author

Abstract: *As a classical audio fingerprint algorithm, the Philips algorithm has been widely used. However, the feature extraction module of the Philips algorithm is inefficient and time-consuming, and the feature binarization algorithm is susceptible to noise interference, which affects the correct rate of audio fingerprint matching. To solve the problems mentioned, this paper proposes an improved Philips audio fingerprint algorithm. The Gammatone filter bank is used to analyze the frequency spectrum of the audio signal to simulate the frequency-selective characteristic of the basilar membrane of the human ear, and a graph-convolutional neural network is introduced to extract the global features between different frequency bands. The distance correlation coefficient is used to calculate the distance between the audio feature matrices to achieve the matching of audio fingerprints. The experimental results show that compared with the original Philips algorithm, the algorithm proposed in this paper achieves lower time consumption and stronger noise immunity.*

Keywords: *Audio matching; Gammatone filter; Graph Convolutional Networks (GCN); Distance correlation*

1. Introduction

The rapid development of the Internet and digital media has led to the emergence of massive amounts of data, and the use of audio in various industries has become more and more extensive. Thus improving the accuracy and real-time performance of audio data processing has become a new research trend. Audio fingerprinting, a technique that extracts the unique features or fingerprints of an audio file and then compares them with existing audio databases to achieve fast identification and retrieval of audio data [1-2], is widely used in many industries, with the most common applications including music identification and broadcast monitoring. Audio fingerprinting can also be used to examine recordings to determine whether their signals have been modified [3].

Because of the increasing use of audio fingerprinting, many algorithms have been proposed, such as the Philips audio fingerprinting algorithm, the Shazam audio fingerprinting algorithm, and a wavelet-based audio fingerprinting algorithm invented by Google [3]. Among these algorithms, the Philips audio fingerprinting algorithm is widely used due to its excellent performance and robustness. However, the Philips audio fingerprint algorithm has the disadvantages of long Bit Error Rate (BER) calculation time and excessive memory space; meanwhile, it is prone to noise interference during binary fingerprint extraction resulting in fingerprint distortion. Therefore, this paper improves the Philips audio fingerprinting algorithm and proposes an audio matching algorithm using the Gammatone filter bank to replace the original Mayer filter bank to improve the ability to capture audio features; introduces Graph Convolutional Networks (GCN) for extracting global features and mining deep information; uses computational distance correlation to calculate the distance between the fingerprints and the binary fingerprints. Deeper information; the distance correlation coefficient is used for matching to improve the anti-interference ability. After experimental verification, the performance of the proposed algorithm is greatly improved compared with the original Philips algorithm.

2. The algorithm proposed in this paper

The flowchart of the improved Philips audio fingerprint algorithm proposed in this paper is shown in Figure 1.

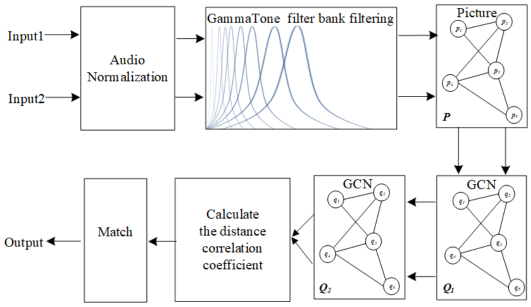
Input1
Input2
Audio Normalization
GammaTone filter bank filtering
Picture
P
GCN
GCN
Q_2
Q_1
Calculate the distance correlation coefficient
Match
Output

Figure 1: Flowchart of the proposed algorithm in this paper.

This algorithm firstly batch normalizes the audio and then filters it using the Gammatone filter bank to obtain its Gammatone Frequency Cepstral Coefficient (GFCC) feature matrix, and then draws a graph and inputs the graph data into the two-layered GCN for processing, and its feature matrix is the GFCC feature matrix, and the adjacency matrix of its plots is the matrix corresponding to the sum of the unit array plus the unit array shifted down one unit.

That is, if a 3-channel Gammatone filter bank is used, the adjacency matrix of the GCN is as follows:

$$\boldsymbol{D}=\begin{bmatrix}1&0&0\\1&1&0\\0&1&1\end{bmatrix} \tag{1}$$

If a 5-channel Gammatone filter bank is used, the adjacency matrix of the GCN is as follows:

$$\boldsymbol{D}=\begin{bmatrix}1&0&0&0&0\\1&1&0&0&0\\0&1&1&0&0\\0&0&1&1&0\\0&0&0&1&1\end{bmatrix} \tag{2}$$

After two layers of GCN computation, the distance correlation is computed between the output feature matrix and the baseline matrix in the database, and the value is subtracted by 1 to obtain the correlation matrix, and the minimum value of each row of the correlation matrix is extracted to compute the sum, which is used as the irrelevance of the song to be matched for the track.

After traversing and calculating all the pre-trained data, the song with the smallest irrelevance is regarded as the song to be matched by the song to be matched. Figure 2 shows the irrelevance of all songs in the library when the first song is without noise.

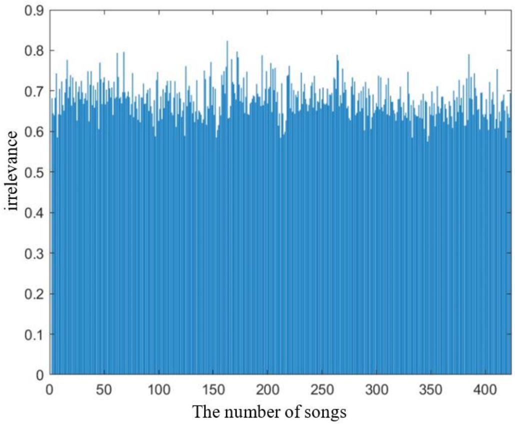

Figure 2: The irrelevance of all songs in the library when the first song is without noise.

The irrelevance to the target song (the first song in the library) is approximately $1.9082\mathrm{e}^{-16}$.

2.1 Audio Normalization

Before the audio data to be processed is fed into the Gammatone filter bank for filtering, the audio data needs to be normalized to facilitate the subsequent operations on the audio data.

Normalization is the process of mapping the original audio data to $[-1,+1]$. In this algorithm, the data to be processed is a column vector, taken as the matrix to be processed, the transpose to facilitate the program processing, for the output matrix, the two mapping relationship is

$$\boldsymbol{y} = \left(\boldsymbol{y}_{max-}\, \boldsymbol{y}_{min}\right) \bullet \frac{\boldsymbol{x} - \boldsymbol{x}_{min}}{\left(\boldsymbol{x}_{max} - \boldsymbol{x}_{min}\right)} + \boldsymbol{y}_{min} \tag{3}$$

After the completion of the operation, the matrix y is transposed to obtain the normalized audio data.

2.2 Gammatone filter bank filtering

The basilar membrane of the cochlea receives external speech signals and decomposes them based on frequency, and the auditory receptor cells are stimulated by travelling wave vibrations. The Gammatone filter bank is a filter model that mimics the frequency decomposition characteristics of the cochlea, whose main function is to decompose the audio signals to facilitate the subsequent feature extraction function. The impulse response of the frequency-centered Gammatone filter bank is

$$g(f,t) = \begin{cases} t^{a-1} e^{-2\pi bt} \cos(2\pi ft), t \geq 0 \\ 0, \text{else} \end{cases} \tag{4}$$

where t represents time; *a* is the filter order, the larger the lower bias; *b* is the time decay coefficient, the larger the shorter the filtering time; *f* is the filter centre frequency [4]. The algorithm uses a total of 32 filters, which have a uniform distribution of centre frequencies on the ERB scale, and filters with higher centre frequencies have a wider bandwidth [5].

The output of the filter preserves the original sampling frequency and its output can be used in many short-duration speech feature extraction algorithms[6]. The amplitude of the down-sampled output is compressed for loudness by a cubic root operation.

$$\boldsymbol{G}_m[i] = \left|\boldsymbol{g}_{\text{dowmsample}}[i,m]\right|^{\frac{1}{3}},\ i = 0,\ldots,N-1, m = 0,\ldots,M-1 \tag{5}$$

where $N = 32$, is the number of channels in the filter bank, *m* is the frame index, and *M* is the number of time ranges obtained later. The generated matrix $\boldsymbol{G}_m[i]$ represents the T-F decomposition of the input[7].

The Gammatone Feature (GF) is inter-correlated due to the overlap of neighboring filter channels. To reduce the dimensionality and eliminate the correlation, this algorithm applies Discrete Cosine Transform (DCT) [8] to the GF. The resulting coefficients are called GFCC [9]. GFCC $\boldsymbol{C}[j], j = 0,\ldots,N-1$

$$\boldsymbol{C}[j] = \sqrt{\frac{2}{N}} \sum_{i=0}^{N-1} \boldsymbol{G}[i] \cos\left[\frac{j\pi}{2N}(2i+1)\right], j = 0,\ldots,N-1 \tag{6}$$

It is stated that the newly derived features are not cepstral coefficients, considering that cepstral analysis requires a logarithmic operation between the first and second frequency analyses for the purpose of back-convolution [10]. The reason for considering these features as cepstral coefficients here is due to the functional similarity between the above transformations and the typical cepstral analysis in the derivation of the Mel Frequency Cepstral Coefficient (MFCC).

2.3 GCN

Graph data, as non-Euclidean spatial data, is capable of representing the structure of our real-life data [11]. GCN deals with graph data, which enables feature extraction, and the parameters of the filters[12] can be shared and convolutional operations can be carried out at all the locations within the graph or at a localized location.

The input to a GCN is a feature description $\boldsymbol{P}$ containing any node i, capable of being written as a feature matrix of $N \cdot D$ (*N* meaning the number of nodes, and *D* the number of input features) and a feature description of the structure of the graph in matrix form, usually in the form of an adjacency matrix.

A GCN model usually outputs $\boldsymbol{Q}$ at the node level (a feature matrix of $N \cdot F$, F meaning the number of output features of any node). At the same time, a introduces a partial pooling operation to the model[13], which is able to do the output at the graph level.

Each neural network layer can be written as a nonlinear function

$$\boldsymbol{H}^{(l+1)} = f\left(\boldsymbol{H}^{(l)}, \boldsymbol{A}\right) \tag{7}$$

In Eq.(7)， $\boldsymbol{H}^{(0)} = \boldsymbol{P}$ ， $\boldsymbol{H}^{(L)} = \boldsymbol{Q}$.

In this formula, $\boldsymbol{Q}$ can also be output as a graph level and L means the number of layers. The key to this model is how the function f is chosen and parameterized.

After integrating the eigenvectors of $\boldsymbol{A}$ and normalizing $\boldsymbol{A}$, we have the following formula

$$f\left(\boldsymbol{H}^{(l)}, \boldsymbol{A}\right) = \sigma\left(\hat{\boldsymbol{D}}^{-\frac{1}{2}} \hat{\boldsymbol{A}} \hat{\boldsymbol{D}}^{-\frac{1}{2}} \boldsymbol{H}^{(l)} \boldsymbol{W}^{(l)}\right) \tag{8}$$

$\boldsymbol{W}$ is the parameter matrix of the l-layer neural network, function σ is the nonlinear activation function, $\hat{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$, $\boldsymbol{I}$ is the unit array, and $\hat{\boldsymbol{A}}$ is the node degree diagonal matrix of the matrix $\hat{\boldsymbol{D}}$.

2.4 Audio Matching Using Distance Correlation Coefficients

The distance correlation coefficient feature selection method provides a novel approach to solving the joint independence problem of testing random vectors[14]. It has good applicability to the relationship between variables, whether linear or nonlinear and is not limited by other parameters or models[15], overcoming the disadvantage that Pearson's correlation coefficient can only measure the degree of linear correlation. At present, the distance correlation coefficient has achieved considerable success in several fields including image recognition[16].

After the GFCC feature matrix has been processed by the GCN, the sample correlation between the GCN output feature matrix and the matrix obtained from the pre-processed audio signal is calculated as 1 minus the sample correlation between the points. If the size of the two matrices is different, the matrices are intercepted with the same size for calculation.

Let the matrix obtained after the processing of the audio signal be matched by $\boldsymbol{K}$, and the reference matrix in the database be $\boldsymbol{S}$. After calculating the sample correlation between the 1-subtracted points, the resulting correlation matrix is.

$$\boldsymbol{Z}_{\text{distance}} = 1 - \frac{\left(\boldsymbol{k}_s - \bar{\boldsymbol{k}}_s\right)\left(\boldsymbol{s}_t - \bar{\boldsymbol{s}}_t\right)'}{\sqrt{\left(\boldsymbol{k}_s - \bar{\boldsymbol{k}}_s\right)\left(\boldsymbol{k}_s - \bar{\boldsymbol{k}}_s\right)'}\sqrt{\left(\boldsymbol{s}_t - \bar{\boldsymbol{s}}_t\right)\left(\boldsymbol{s}_t - \bar{\boldsymbol{s}}_t\right)'}} \tag{9}$$

In Eq.(9)， $\bar{\boldsymbol{k}}_s = \frac{1}{n}\sum_i \boldsymbol{k}_{si}$ ， $\bar{\boldsymbol{s}}_t = \frac{1}{n}\sum_j \boldsymbol{s}_{tj}$.

The irrelevance value between two matrices is considered as the irrelevance value between two matrices by finding the sum of the minimum values in each row of the matrix (https://github.com/yashdv/Speech-Recognition). After iterative computation of all the stored matrices, the result with the smallest irrelevance value is taken and the song corresponding to that matrix is regarded as the matching result.

3. Experimental Analysis

Since the performance of the computer and test parameters have a great impact on the performance of the algorithm, the experimental parameters and test environment are described in Table 1.

Table 1: Experimental parameters and test environment.

Items	Equipment or Settings
Audio sampling rate	8000Hz
Audio length	5sec
Audio sampling accuracy	32bit
Number of the Gammatone filter bank channels	32
Number of GCN delays	2
Total number of test songs	423
Type of additive noise	White Gaussian Noise
The language used by the Philips algorithm	Python
The language used by the improved algorithm	MATLAB
Test computer	Legion Y9000P2021H (Laptop)
CPU	i7-11800H
Memory size	15.8GB
Video card	Intel(R) UHD Graphics

3.1 The comparison of performance between the improved algorithm and the Philips algorithm

The Philips algorithm defines a matching failure as a bit error rate of more than 35%. However, the improved algorithm proposed in this paper doesn’t set any limitation to define a matching failure, so the restriction of the Philips algorithm is removed during the following test.

To prevent subtle errors caused by the randomness of adding noise, the test audio input of the two algorithms is the data after adding noise (Signal-to-Noise Ratio (SNR) is 0dB) and normalization, and the benchmark data is also subjected to the same normalization pre-processing.

Due to the slow operating efficiency of the Philips algorithm, the audio data was reduced to 50 songs. Table 2 shows the test results of accuracy, and Table 3 shows the results of time-consuming.

Table 2: Test results of the two algorithms with an SNR value of 0dB.

	The Philip algorithm	The improved algorithm
The number of correct match songs	2	43
Accuracy/%	4	86

Table 3: Time consumption of the two algorithms.

	The Philip algorithm	The improved algorithm
The time-consuming of calculating a song's feature matrix /sec	about 3.8	about 0.067
The time consuming of calculating the similarity or bit error rate of two songs /sec	about 0.007	the first song is about 0.11, other songs are about 0.013

The Philip algorithm has the problem that the matching rate is too low in a high-noise environment, the space utilization rate is also low when calculating audio, and the memory consumption is too large, resulting in the algorithm time being too long [1].

Experimental data show that compared with the Philips algorithm, the improved algorithm has a significant improvement in matching accuracy, especially in the case of low SNR, and the time consumption is far less than the Philips algorithm.

In the improved algorithm, the Mel-filter bank is changed to the Gammatone filter bank, and GCN is introduced to extract the global feature information, which greatly improves the performance of the algorithm. The algorithm changes the matching method to calculate the distance correlation, which improves the anti-noise ability and operation efficiency of the algorithm.

- ***The anti-noise ability test of the algorithm in this paper***

The anti-noise ability of the algorithm is a very important indicator to measure the excellence of the algorithm so this paper tests the matching accuracy of the algorithm after adding noise under different SNR. The test results of the number of successful matches of songs under different SNR noises are shown in Table 4. To reflect its changing trend, Figure 3 shows the change in the number of successful matches of songs with the increase of SNR.

Table 4: Test results under different SNR.

SNR/dB	0	5	10	15	20
The number of correct match songs	388	412	422	423	423
Accuracy/%	91.73	97.40	99.76	100.00	100.00

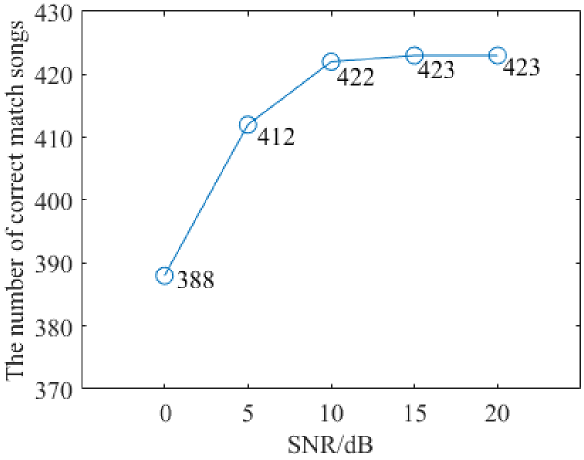

Figure 3: Match number after adding different noises.

Experiments show that the improved algorithm has good anti-noise ability. When the SNR is greater than or equal to 10dB, the algorithm matching accuracy is more accurate. When the SNR is less than 10dB, the algorithm performance will rapidly decline with the increase in noise intensity.

When testing the performance of the algorithm, the audio data sampling rate is too low (8000Hz) and the length is too short (5sec), which greatly increases the difficulty of matching audio, especially in the case of low SNR. In addition, the experimental data set contains some similar audio clips, which puts forward higher requirements for the performance of the algorithm.

When the SNR is equal to 0dB, the algorithm is close to the performance limit, by calculating the difference value in the similarity between the song to be matched and other songs with the least similarity in the database and the target song, and setting the difference value as R. Figure 4 shows how the R-values of the first and second songs change with SNR.

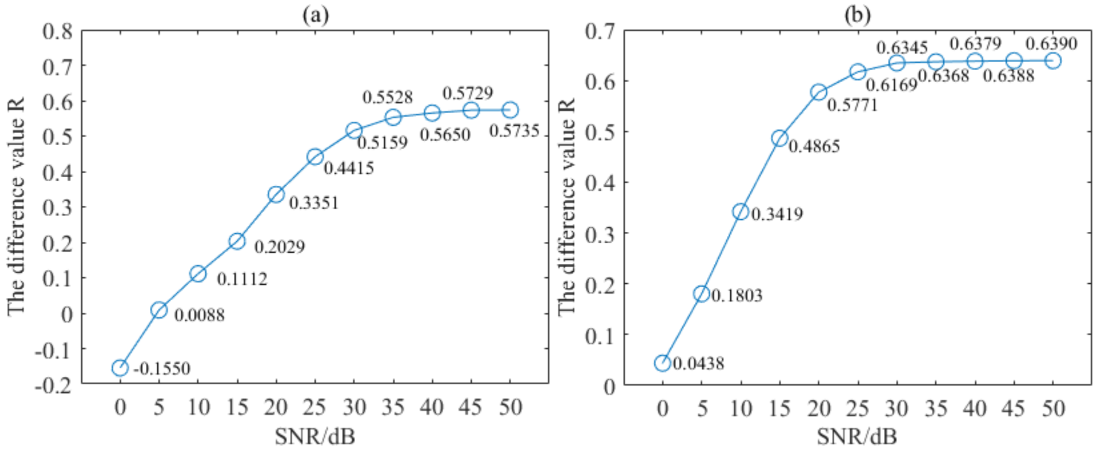

Figure 4: Variation of R-value of songs with different SNR: (a) the first song; (b) the second song.

Experiments show that in the low SNR environment of the improved algorithm, songs with a high degree of similarity may lead to matching errors, so the similarity of songs will also have a great impact on the performance of the algorithm.

- ***Performance Comparison of Gammatone Filter Banks vs. Mel Filter Banks***

The experiment replaces the Gammatone filter bank with a Mayer filter bank and ensures that the intercepted matrices used to calculate the distance correlation coefficients are of equal size under the different filter banks. The test results are shown in Table 5.

Table 5: Test results under different Filter Bank.

	Mel-filter bank	Gammatone filter bank
The number of correct match songs	198	373
Accuracy/%	46.81	88.18

The experiments show that the use of the Gammatone filter bank has a great improvement in the performance of the algorithm with better noise immunity compared to the use of the Mel filter bank, improves the algorithm's noise immunity at low SNR, and reduces the probability of collision between different audios.

3.2 Effect of different parameters on the performance of this algorithm

This algorithm contains several parameters, and the effects of several major parameters on the performance of the algorithm will be analyzed next.

- ***Effect of the number of channels in the filter bank***

In this experiment, several different parameters are selected to test the effect of the number of channels in the filter bank on the performance of the algorithm. The selection of parameters and test results are shown in Table 6.

Table 6: Test results under different channel numbers.

Number of channels used to test the sample	Number of channels used to train benchmark sample	The number of correct match songs	The number of correct match songs
16	16	386	91.25
32	16	386	91.25
16	32	376	88.89
32	32	388	91.73
64	32	389	91.96
128	32	386	91.25
32	64	388	91.73
64	64	386	91.25
32	128	385	91.02
128	128	384	90.87

To further reflect its influence, Figure 5 respectively shows the test results when the number of base sample channels is unchanged, the number of test sample channels is unchanged, and the number of channels in the two samples is changed simultaneously.

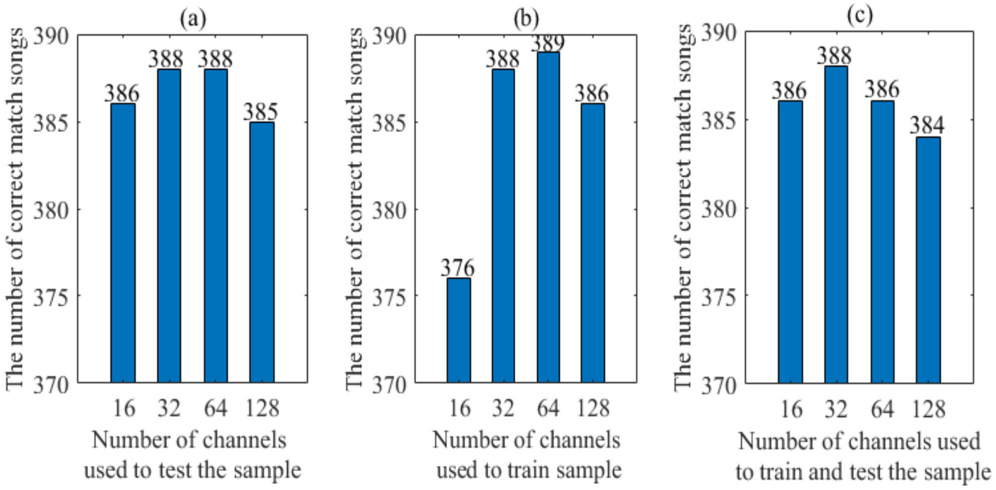

Figure 5: Test results under different channel numbers:(a) Keep the number of base sample channels to 32; (b) Keep the number of test sample channels to 32; (c) Change the number of channels at the time.

Experiments have shown that the algorithm is not greatly affected when changing the number of channels in the filter bank of the test samples or the number of channels in the filter bank of the base samples individually, and its effect is negligible.

In the process of experimentally intercepting the feature matrix, more feature information is lost in the high-frequency part, while the impact on the low-frequency part is relatively small. This algorithm is more compatible for different numbers of channels.

The experiments also show that when the number of channels in the filter bank is too small, the algorithm is less resistant to interference and performs poorly due to few features extracted (i.e., the feature matrix is too small), and the collision frequency increases; when the number of filter channels is too large, there is a decrease in the frequency range of features extracted from the channels, and at the same time, due to the asymmetric nature of the Gammatone filter bank, the impact caused by the noise may increase, which further decreases the accuracy of the feature extraction and results in an audio matching error.

- ***Effect of the number of GCN layers***

Since the number of layers of GCN will affect the performance of the algorithm, this section will discuss the effect of the number of layers on the performance of the algorithm by testing different. The test results under different numbers of GCN layers are shown in Table 7.

Table 7: Test results with different numbers of GCN layers.

Number of GCN layers	0	1	2
The number of correct match songs	373	386	388
Accuracy/%	88.18	91.25	91.73

Experiments show that the addition of GCN links has a significant improvement on the song-matching success rate, and the number of successfully matched tracks increases from 373 to 388 when using two layers of GCN. The experiments show that GCN achieves more accurate global information extraction, but there is still some room for improvement.

Meanwhile, in the experiment, two-layer GCN has some improvement in matching the correct rate compared with one-layer GCN, but the improvement space is small. It is analyzed that this is related to the excessive over-smooth problem of GCN. Since the amount of data is too small and the value of each node has already converged or converged, adding another layer of GCN will not cause too much impact on the performance. Further improvement of the algorithm can also be achieved by trying to improve the transition over-smooth problem.

- ***Impact of matching method***

This experiment aims to improve the performance of the algorithm by changing the matching method and comparing i with the method of calculating Euclidean distance. The test results are shown in Table 8.

Table 8: Test results under different matching methods.

Matching Method	Euclidean distance	Distance correlation
The number of correct match songs	1	388
Accuracy/%	0.00	91.73

Experiments have shown that calculating the distance correlation coefficient for matching has greatly improved the performance of the algorithm. The advantage of the distance correlation coefficient is that it can determine the nonlinear correlation between two variables, overcoming the disadvantage of the Pearson correlation coefficient, which can only measure the linear relationship.

3.3 Algorithm Performance Evaluation

The strength of the interference has a great impact on the performance of the algorithm, and since the impact of noise on the performance of the algorithm in this paper has been analyzed in the previous section, the impact of the audio length on the performance of the algorithm will be analyzed mainly here.

- ***Effect of audio length***

Since audio length differences can cause the final feature matrix size to be different from the baseline matrix in the database, the same matrix size will be intercepted to calculate the distance correlation coefficients and irrelevance when performing this experiment. Table 9 and Table 10 show the matching accuracy with audio length when no noise is added and when SNR is 0, respectively. To illustrate the effect of audio length on accuracy, Figure 6 shows the variation trend of accuracy with audio length when SNR is 0.

Table 9: Test results without any noise.

Audio length /sec	1	2	3	4	5
the number of correct match songs	422	423	423	423	423
accuracy/%	99.76	100.00	100.00	100.00	100.00

Table 10: Test results with an SNR value of 0dB.

Audio length /sec	1	2	3	4	5
The number of correct match songs	183	314	357	374	388
Accuracy/%	43.26	74.23	84.40	88.42	91.73

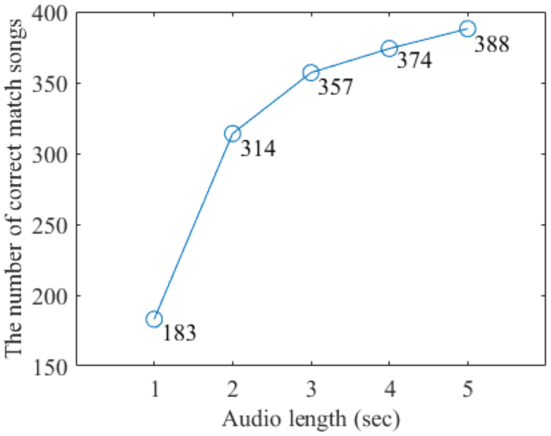

Figure 6: Test results with an SNR value of 0dB.

Experiments show that the algorithm has a good anti-interference ability, but when the input audio length is too short, it causes a more obvious performance degradation due to a sudden increase in the collision probability caused by too few extracted features. At the same time, the sampling rate of the test sample is only 8000Hz, which will further reduce the number of extracted features and the degree of accuracy, greatly reducing the size and accuracy of the matrix used for matching, resulting in an increase in the probability of collision and reducing the correct matching rate.

The asymmetry of the Gammatone filter bank results in more information being extracted at lower frequencies, which better simulates the human ear's response to sound, while the distance correlation coefficient accurately measures the degree of nonlinear correlation between the two samples, further overcoming the effects of audio shortening and improving matching compatibility.

4. Conclusions

It is proved experimentally that the algorithm proposed in this paper has great improvement compared to the Philips algorithm, both in terms of time efficiency and matching success rate. In the process of testing the GCN, the difference in drawing methods also greatly affects the performance of the GCN, and thus the improvement of the algorithm can start from here. Meanwhile, due to the over-smooth problem of GCN, further improvement of the algorithm can consider replacing GCN with a depth map convolutional neural network or other solutions. This paper does not discuss the case where the song is not in the database and focuses on researching and improving the anti-jamming performance of the algorithm, and more in-depth research can be carried out from that aspect as well, by introducing a new judgment metric for that case.

References

[1] Sun N, Zhao W, Chen M, et al. An Improved Algorithm of Philips Audio Fingerprint Retrieval [J].Computer Engineering, 2018, 44 (1): 280-284.

[2] Balado F, Hurley N J, McCarthy E P, et al. Performance of philips audio fingerprinting under additive noise[C]//2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07. IEEE, 2007, 2: II-209-II-212.

[3] Wei Xiong, Xiaoqing Yu, Jianhua Shi. An improved audio fingerprinting algorithm with robust and

efficient[C]//IET International Conference on Smart and Sustainable City 2013 (ICSSC 2013). Shanghai, 2013. DOI: 10.1049/cp.2013.1960.

[4] Patterson, Roy D., Walters, Thomas C., Monaghan, Jessica, et al. Auditory speech processing for scale-shift covariance and its evaluation in automatic speech recognition[C]. //Proceedings of 2010 IEEE International Symposium on Circuits and Systems: ISCAS 2010, Paris, France, 30 May - 2 June 2010, Pages 1-736, [v.1]. :IEEE, 2010:3813-3816.

[5] Moore B C J. An introduction to the psychology of hearing[M]. Brill, 2012.

[6] Ranjan R, Thakur A .Analysis of Feature Extraction Techniques for Speech Recognition System[J]. International Journal of Innovative Technology and Exploring Engineering, 2019.

[7] Wang D L, Brown G J. Computational auditory scene analysis: Principles, algorithms, and applications [M]. Wiley-IEEE press, 2006.

[8] Zhao X, Shao Y, Wang D L .CASA-Based Robust Speaker Identification [J]. IEEE Transactions on Audio Speech and Language Processing, 2012, 20(5):1608-1616. DOI:10.1109/TASL.2012.2186803.

[9] Zhao X, Shao Y, Wang D L. Robust speaker identification using a CASA front-end[C]//2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2011 :5468-5471. DOI:10.1109/ICASSP.2011.5947596.

[10] Shao Y, Wang D L .Robust speaker identification using auditory features and computational auditory scene analysis[C]//IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2008.DOI:10.1109/ICASSP.2008.4517928.

[11] Hu K, Wu J, Li Y, et al. FedGCN: Federated Learning-Based Graph Convolutional Networks for Non-Euclidean Spatial Data[J].Mathematics, 2022, 10.

[12] Duvenaud D, Maclaurin D , Aguilera-Iparraguirre J ,et al.Convolutional Networks on Graphs for Learning Molecular Fingerprints[J].MIT Press, 2015.DOI:10.48550/arXiv.1509.09292.

[13] Wang C, Pan S, Long G, et al. Mgae: Marginalized graph autoencoder for graph clustering[C]//Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. 2017: 889-898.

[14] Székely, Gábor J, Rizzo M L , Bakirov N K. Measuring and testing dependence by correlation of distances. [C]//ACM Symposium on Virtual Reality Software & Technology. ACM, 2007. DOI:10.1145/1315184.1315209.

[15] Ngo S H, S. Kemény,A. Deák.Performance of the ridge regression method as applied to complex linear and nonlinear models[J].Chemometrics & Intelligent Laboratory Systems, 2003, 67(1):69-78. DOI:10.1016/S0169-7439(03)00062-5.

[16] Zhen X, Meng Z, Chakraborty R, et al. On the versatile uses of partial distance correlation in deep learning [C]//European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2022: 327-346.