ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 2 - 2026

Research on Deep Learning-Based Vocal Separation and Digital Synthesis in Choral Music

Luanyu Zhao[1,*]

[1] Department of International Studies, Chugye University for Arts, Seodaemun-gu, Seoul, 03762, Korea

SUMMARY: *With the advancement of computer performance, audio processing technology has also achieved significant development. This study explores the application of deep learning in audio separation and music synthesis, introducing deep recurrent neural networks into audio separation to construct a choral voice separation model. It further proposes a choral vocal synthesis system encompassing both training and synthesis, built upon deep neural networks. Experimental results demonstrate that the DNN-based choral voice separation algorithm effectively isolates vocals from accompaniment. The separated choral vocals achieve a significantly higher PESQ score of 3.32 compared to both the original track and accompaniment, outperforming comparison algorithms by 26.82% to 35.10%. The DNN model achieved musical metrics close to the original, with errors under 10%. Its generated choral music demonstrated lower TER values across all vocal parts compared to other algorithms, remaining within 5%, indicating excellent choral vocal synthesis performance. This method enables high-quality vocal separation and generation, showcasing its potential applications in music creation.*

KEYWORDS: *Deep learning; Deep recurrent neural network; Deep neural network; Sound separation; Vocal synthesis*

1 Introduction

Audio serves as humanity's most familiar medium for conveying messages, intentions, and emotions, characterized by its capacity to carry substantial, nuanced, and precise information [1-3]. With advancements in multimedia information processing technologies and enhanced computational capabilities, audio processing techniques have gained significant attention and widespread application [4, 5]. In recent years, the introduction of deep learning methods has substantially propelled progress in speech technology [6]. Deep learning technology, by constructing multi-layer neural network structures, can automatically learn high-level features from unstructured data [7, 8]. As practical applications like sound separation, digital synthesis, and speech recognition increasingly permeate daily life, deep learning-based speech technologies have become a crucial component of artificial intelligence and a primary direction for real-world implementation [9, 10].

Currently, speech separation is a highly active area in signal processing, with a substantial research foundation established [11]. Common methods include Bayesian models, Hidden Markov Models (HMMs), and Gaussian Mixture Models (GMMs). For instance, [12] introduced a general Bayesian source model framework and conducted experimental studies on

*zhaoluanyu8@163.com
https://doi.org/10.65102/is2026643
Published: 30 April, 2026

separating vocals from music. Results demonstrated consistent vocal separation with superior performance. Reference [13] optimized speech enhancement technology based on hidden Markov models to achieve speech separation. Validation experiments conducted on a small-vocabulary audiovisual database demonstrated significant improvements in machine interpretability.

Deep learning-based automatic sound source separation methods can isolate overlapping audio signals, effectively enhancing speech recognition efficiency and accuracy [14]. For instance, [15] jointly optimizes deep learning models with masking layers to evaluate band-pass sound source separation performance. Compared to reference models, the optimized model achieves a 3.8–4.9 dB improvement in signal-to-noise ratio and demonstrates stronger signal enhancement rates. Reference [16] proposes a data-driven deep learning-based method for separating underwater acoustic signals. This approach achieves effective separation even under 40 dB Gaussian noise, with an average similarity coefficient exceeding 0.8 in high-noise conditions. Reference [17] introduced a novel deep learning model to evaluate its performance in separating neonatal chest sounds. This method achieved over 17 times faster separation speed than the baseline model, with measured sound separation distortion reduced by 2.01 to 5.06 dB.

Speech synthesis is a technology that converts text into corresponding speech through computer processing of textual information [18]. Reference [19] utilized the XiaoiceSing digital singing voice synthesis system, achieving vocal quality scores 1.44 dB higher than the convolutional neural network baseline model, pitch accuracy 1.18 dB higher, and naturalness 1.38 dB higher, while securing over 80% preference rates. In recent years, driven by AI advancements, deep learning-based speech synthesis network models have been developed and effectively applied. Reference [20] employed the E2E-V2SResNet model based on a convolutional encoder-decoder framework to accurately synthesize speech from the GRID database. Its performance surpassed baseline models in both speech quality and intelligibility, achieving a 3.077% improvement in speech quality and a 2.593% increase in speech intelligibility. Reference [21] employs a text-to-speech synthesis system based on deep neural networks to model the relationship between musical scores and their acoustic features. The synthesized singing voice demonstrates superior naturalness compared to alternative methods, validating the effectiveness of deep learning networks in voice synthesis. Reference [22] utilizes convolutional neural networks to model long-term dependencies in singing voices. Combined with techniques to reduce computational complexity, this approach achieves significantly faster synthesis speeds than traditional methods while producing more natural vocal expressions.

The growing demand for diverse choral art forms has been met through integrated applications of voice separation and intelligent synthesis technologies, significantly enhancing production efficiency and content quality in the arts industry. Innovative applications of deep learning in personalized recommendations, sentiment analysis, and interactive entertainment provide new momentum for its further development [23, 24].

Based on an analysis of audio separation and digital synthesis applications, this research employs a deep recurrent neural network (DNN) to account for musical contextual information. Leveraging its multi-layer nonlinear structure, the DNN learns optimal latent representations. These learned representations are then used to reconstruct both the vocal and instrumental parts. A joint optimization of the neural network and the use of soft template functions enables the separation of these components, achieving a DNN-based choral vocal separation model. Subsequently, an acoustic model with music-to-acoustic feature mapping functionality is trained using DNN, augmented with vibrato and delay models to generate chorus vocals that closely resemble natural singing. Finally, a chorus song database is compiled as a delay database. Using PESQ as the evaluation metric, the audio separation effectiveness of the DRNN chorus

voice separation model is explored. Additionally, from both musicality and model performance perspectives, the musicality metrics and model comparison results of the DNN model are analyzed to validate the musical quality of the generated chorus vocals and the model's performance.

2 Applications of Audio Separation and Digital Synthesis

2.1 Applications of Audio Separation Technology

The application value of audio separation technology in music manifests in multiple aspects, with its practical significance most notably demonstrated in three fields: auditory training, music composition, and audio restoration. Driven by the rapid advancement of artificial intelligence, this technology has evolved into a comprehensive, powerful, and intelligent technological system. In the following sections, this study will integrate deep learning methods to construct a choral voice separation approach.

2.1.1 Auditory Training

(1) Enhancing Auditory Discrimination Skills

Audio separation technology enables the decomposition of complex mixed audio into distinct audio track sources. This not only deconstructs multi-part musical layers vertically but also stretches their tempo horizontally, thereby improving listeners' comprehension and identification of various musical works. Simultaneously, by comparing the isolated parts, students can more precisely perceive subtle changes in musical detail and emotional control, thereby enhancing their overall musical expressiveness.

(2) Expanding the Depth and Breadth of Auditory Training Materials

On one hand, this broadens the scope and diversity of auditory training resources. On the other, it increases course engagement while enabling the direct use of complex, cutting-edge musical materials in current practice, making students more willing to participate and invest in training.

(3) Facilitating Personalized Ear Training Instruction and Adaptive Learning Models

On one hand, instructors can tailor auditory training to each learner's specific needs and proficiency level. On the other hand, students can utilize this technology for self-directed practice and operation, which not only enhances their self-assessment and adjustment capabilities but also cultivates habits and motivation for independent learning.

2.1.2 Music Composition

(1) Unlocking Diverse Creative Inspiration Sources

Audio separation technology enables music creators to directly extract specific tracks or sound materials from existing works, repurposing them as new creative elements or background sounds.

(2) Enhancing Music Remixing and Arrangement Capabilities

During music composition, separation tools can extract multiple vocal parts from original works for remixing, allowing different instruments or rhythms to be rearranged. Simultaneously, parameters like volume, frequency, and effects can be adjusted and reconfigured to create entirely new musical styles and auditory experiences.

(3) Improving Audio Source Material Management Efficiency

After separating complex audio, individual tracks can be tagged with more precise metadata, such as specific genres, instrument names, vocal characteristics, and rhythm types. Detailed database classification management enhances audio material retrieval efficiency, simplifying

editing and processing workflows for creators. This enables rapid sourcing of required assets, facilitating better organization and access to existing musical resources.

2.1.3 Audio Restoration

(1) Precision Restoration of Damaged Audio

Audio restoration represents a critical application of audio separation technology. By isolating specific elements within an audio track, damaged sections can be pinpointed with greater accuracy, enabling targeted restoration and repair.

(2) Substantial Enhancement of Audio Quality

By isolating and enhancing specific sound sources, operations such as noise reduction, echo cancellation, volume balancing, frequency response adjustment, and stereo width optimization are performed. This results in audio that is more forward-projecting, clear, full-bodied, and smooth.

(3) Customized Audio Restoration Solutions

Each audio clip presents unique restoration requirements and challenges. Audio separation technology enables the creation of tailored restoration solutions based on the specific circumstances of each case.

2.2 Applications of Digital Music Synthesis

Digital music synthesis refers to the process of automatically generating musical fragments using electronic devices, primarily leveraging artificial intelligence technologies within digital systems. Most common AI music software today relies on machine learning algorithms and neural networks. By analyzing existing musical data, these systems can generate entirely new musical compositions. This paper primarily employs deep learning methods using DNNs to investigate techniques for synthesizing vocal sounds.

In traditional music creation workflows, composers typically develop the melody part first, followed by the accompaniment parts. Creating a song involves multiple steps such as lyric writing, composition, recording, and mixing, making the process relatively complex. However, the use of digital music generation software can significantly improve this situation, greatly enhancing the efficiency of music fragment creation.

From a practical application perspective, while digital music generation enhances efficiency and diversifies creative tools, it also presents composers with numerous challenges. For instance, during music generation, composers must ensure musical fragments adhere to established rules to maintain consistency between melodic and textural patterns. Disconnected notes may violate listeners' auditory expectations, leading to ambiguous tonality. Therefore, creators must synthesize insights while operating these tools to grasp the intrinsic logic of melody and refine overall quality through subtle adjustments.

3 Choral Voice Separation and Digital Synthesis Methods

3.1 DRNN-Based Vocal Separation in Choral Music

This section applies deep recurrent neural networks (DRNNs) to the field of music separation. By leveraging the neural network's powerful learning and classification capabilities, it models the complex structure of choral audio signals to achieve superior separation results.

3.1.1 DRNN

Deep recurrent neural networks differ significantly in structure from earlier feedforward neural networks, yet they share the same underlying principles.

Recurrent neural networks are predominantly applied to time-series data. They consist of an input layer x, a hidden layer s (also known as the context layer or state layer), and an output layer y At time step t, the input is $x(t)$, the output is $y(t)$, and the state layer is $s(t)$. The input vector $x(t)$ is the sum of the current state vector w and the previous state layer output $s(t-1)$, as follows:

$$x(t) = w(t) + s(t-1) \tag{1}$$

$$s_j(t) = f\left(\sum_i x_i(t) u_{ji}\right) \tag{2}$$

$$y_k(t) = g\left(\sum_j s_j(t) v_{kj}\right) \tag{3}$$

Here, $f(z)$ is the sigmoid activation function:

$$f(z) = 1/(1+e^{-z}) \tag{4}$$

Here, $g(z)$ is the softmax function:

$$g(z_m) = e^{z_m} / \sum_k e^{z_k} \tag{5}$$

The greatest advantage of deep recurrent neural networks is their ability to utilize contextual information. To obtain contextual information from musical signals, the common approach is to concatenate neighboring features as input to the deep neural network. However, this significantly increases the number of parameters based on the input dimension. Therefore, it is necessary to limit the size of the window connecting adjacent features. Recurrent neural networks (RNNs) can be viewed as deep neural networks with an indefinite number of layers, enabling them to retain memory from previous time steps. The primary limitation of RNNs is their inability to process complex inputs at the current time step. To handle intricate information across different temporal scales, this paper employs Deep Recurrent Neural Networks (DRNNs).

Here we define different types of DRNNs. Consider a DRNN with l hidden layers, where the l th layer is recurrently connected. The activation function for the l th hidden layer at time t is defined as:

$$h_t^l = f_h(x_t, h_{t-1}^l) = \phi_l(U^l h_{t-1}^l + W^l \phi_{t-1}(W^{l-1}(..\phi_1(W^1 x_t)))) \tag{6}$$

The output y_t can be defined as:

$$y_t = f_o(h_t^l) = W^L \phi_{L-1}(W^{L-1}(...\phi_l(W^l h_t^l))) \tag{7}$$

where x_i is the input to the neural network at time t, ϕ_i is a dot-product nonlinear function, w^l is the weight matrix for layer l, $U^{'}$ is the weight matrix for the recurrent connections in layer l, and the output layer is a linear layer.

Block RNNs consist of multiple layers of transfer functions, defined as:

$$h_t^l = f_h(h_t^{l-1}, h_{t-1}^l) = \phi_l(U^l h_{t-1}^l + W^l h_t^{l-1}) \tag{8}$$

Here, h_i^l denotes the hidden state at layer l at time t. U^l and W^l represent the weight matrices for the hidden activation function and the lower-layer activation h_i^{l-1} at time $t-1$, respectively. When $l=1$, the hidden activation function can be computed as $h_i^0 = x_i$.

The function $\phi_i(\cdot)$ is a nonlinear function. This paper employs the modified linear unit function $f(x) = \max(0, x)$, which yields better results than sigmoid and tanh functions. For a deep neural network, the weight matrix U^l is initially set to a zero matrix.

3.1.2 Model Framework

At time t, the training input x_l for the neural network is a feature window of the mixed signal, typically represented by combining multiple frames. In this chapter, the amplitude spectrum is used as the feature. The output targets are y_{1t} and y_{2t}, while the output predictions are $\hat{y}_{1t}$ and $\hat{y}_{2t}$, both representing amplitude spectra from different sources.

The music separation framework based on deep recurrent neural networks, as shown in Figure 1, not only incorporates phase spectrum information but also employs discriminative training.

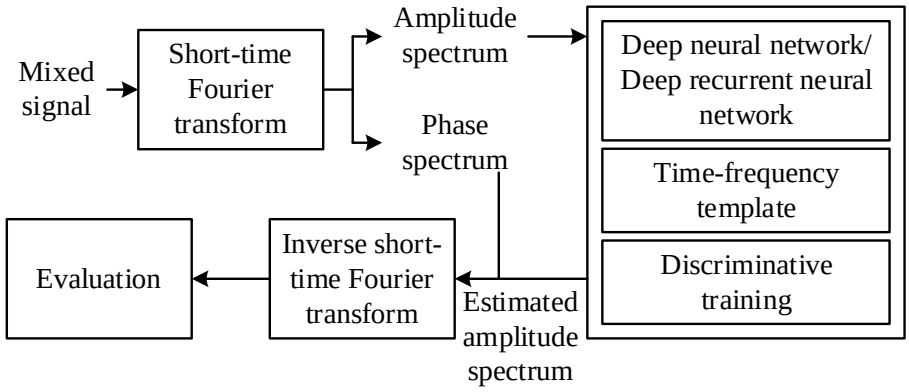

Figure 1: Music separation Framework Based on deep recurrent neural networks

3.1.3 Time-Frequency Analysis

Given the input feature x_i of a mixed signal, the network then produces output predictions $\hat{y}_{1t}$ and $\hat{y}_{2t}$. The soft template m_i is defined as follows:

$$m_t(f) = \frac{|\hat{y}_{1t}(f)|}{|\hat{y}_{1t}(f)| + |\hat{y}_{2t}(f)|} \tag{9}$$

Here, f represents different frequency components.

Once a time-frequency template m_t is computed, it is then applied to the amplitude spectrum z_t of the mixed signal to obtain the estimated separated spectra $\hat{s}_{1t}$ and $\hat{s}_{2t}$ corresponding to sources 1 and 2, respectively, as expressed below:

$$\hat{s}_{1t}(f) = m_t(f) z_t(f) \tag{10}$$

$$\hat{s}_{2t}(f) = (1 - m_t(f)) z_t(f) \tag{11}$$

Here, f represents different frequency components.

The time-frequency masking function can be viewed as a layer within the neural network. This chapter employs joint training to optimize both the neural network and the time-frequency masking function, effectively adding a layer as the output of the original neural network.

$$\hat{y}_{1t} = \frac{|\hat{y}_{1t}|}{|\hat{y}_{1t}| + |\hat{y}_{2t}|} \odot z_t \tag{12}$$

$$\hat{y}_{2t} = \frac{|\hat{y}_{2t}|}{|\hat{y}_{1t}| + |\hat{y}_{2t}|} \odot z_t \tag{13}$$

Here, $\odot$ denotes the Hadamard product. The entire neural network is constrained, and the algorithm is jointly optimized in conjunction with the masking function. The added extra layer is a decision layer, where the neural network weights are updated via backpropagation after calculating the errors between $\hat{y}_{1t}, \hat{y}_{2t}$ and y_{1t}, y_{2t}. To achieve smoother predictions, the previous template function is employed to estimate $\hat{s}_{1t}$ and $\hat{s}_{2t}$. Subsequently, the separated source signals are obtained based on the phase spectrum of the ISTFT transform and the original signal.

3.1.4 Discrimination Training

Given the original signals y_{1t}, y_{2t} and the output predictions $\hat{y}_{1t}, \hat{y}_{2t}$, the parameters of the neural network are then optimized by minimizing the mean squared error and the generalized Kullback-Leibler divergence, with the mathematical expressions as follows:

$$J_{MSE} = \| \hat{y}_{1t} - y_{1t} \|_2^2 + \| \hat{y}_{2t} - y_{2t} \|_2^2 \tag{14}$$

$$J_{KL} = D(y_{1t} | \hat{y}_{1t}) + D(y_{2t} | \hat{y}_{2t}) \tag{15}$$

Here, $D(A|B)$ is defined as follows:

$$D(A|B) = \sum_i (A_i \log \frac{A_i}{B_i} - A_i + B_i) \tag{16}$$

The condition for $D(\cdot|\cdot)$ to become the K-L divergence is $\sum_i A_i = \sum_i B_i = 1$, where A and B are treated as probability distributions.

However, to enhance similarity between prediction and target, minimize equations (14) and (15). One objective of source separation is to achieve a higher SIR value. Here, we seek a discriminative objective function not only to increase similarity between prediction and target but also to reduce similarity between the target and other sources, expressed as follows:

$$\|\hat{y}_{1t} - y_{1t}\|_2^2 - \gamma\|\hat{y}_{1t} - y_{2t}\|_2^2 + \|\hat{y}_{2t} - y_{2t}\|_2^2 - \gamma\|\hat{y}_{2t} - y_{1t}\|_2^2 \tag{17}$$

$$D(y_{1t} \mid \hat{y}_{1t}) - \gamma D(y_{1t} \mid \hat{y}_{2t}) + D(y_{2t} \mid \hat{y}_{2t}) - \gamma D(y_{2t} \mid \hat{y}_{1t}) \tag{18}$$

Here, γ is a performance constant on the dev test set.

In summary, the main steps of a deep learning-based music separation algorithm are as follows:

First, perform a short-time Fourier transform on the mixed signal, then feed these as input to a deep recurrent neural network. Next come the two crucial steps of the deep learning-based separation algorithm:

(1) Training phase

Using the spectrum as input, the forward algorithm continuously calculates outputs. These are compared with the true outputs, and the objective function is optimized using gradient descent until iteration stops. This yields the weights, biases, and other parameters of the deep recurrent neural network.

(2) Separation Phase

The network parameters obtained during training are applied to separate the test set. Evaluation metrics are then used to derive individual parameters, and the average is calculated across the entire test set. Finally, by adjusting these parameters, the average values for other test sets can be recalculated.

3.2 DNN-Based Vocal Synthesizer for Choral Music

3.2.1 Vocal Synthesis Framework

Deep neural networks (DNNs), also known as multilayer perceptrons (MLPs), are discriminative models that learn the distribution of training samples through supervised learning, subsequently performing classification or regression tasks. Based on layer positioning, the neural networks within a DNN are categorized into three types: input layer, hidden layer, and output layer. The model may have multiple inputs and outputs, with the intermediate hidden layer potentially comprising multiple layers, all interconnected through fully connected connections. The DNN-based choral singing synthesis system, as shown in Figure 2, consists of two main components: training and synthesis.

The training component extracts parameters such as fundamental frequency and Mel spectrum from recorded vocal samples in the database. These parameters are used to train the DNN, yielding an acoustic model based on the DNN. Since DNN modeling primarily relies on speech acoustic parameters, its input and output features require temporal alignment. Therefore, the input and output features for the DNN here are the results of frame-by-frame temporal alignment performed by a trained HMM. The synthesis component first analyzes the musical score of the target vocal track to derive context-dependent annotation sequences. These annotation sequences are then mapped to acoustic feature sequences via forward propagation through the trained DNN acoustic model, generating spectral parameters and excitation signals. Finally, the vocal track is synthesized using an acoustic codec.

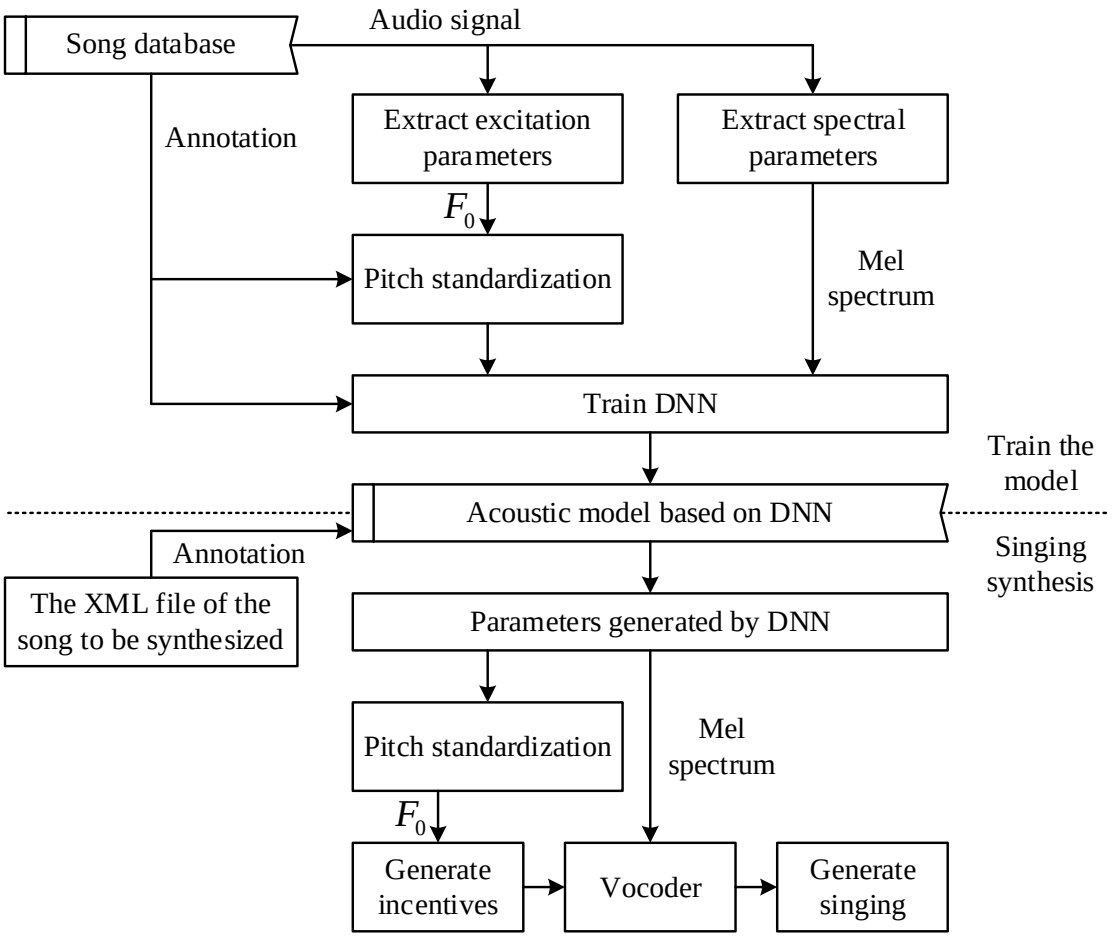

Figure 2: The chorus music synthesis of DNN

3.2.2 Trajectory Training

When synthesizing choral vocals using DNNs, training an acoustic model with the capability to map musical features to acoustic features requires modeling both static and dynamic characteristics within the DNN to generate smooth parameter trajectories.

The acoustic feature vector o_t is a vector containing both D-dimensional static features $c_t = [c_t(1), ..., c_t(D)]^T$ and dynamic features:

$$o_t = [c_t^T, \Delta 1 ta^{(1)} c_t^T, \Delta 1 ta^{(2)} c_t^T]^T \tag{19}$$

The acoustic feature vector o and static feature vector c of a song sequence can be represented as:

$$o = [o_1^T, ..., o_t^T, ... o_T^T]^T \tag{20}$$

$$c = [c_1^T, ..., c_t^T, ... c_T^T]^T \tag{21}$$

where T is the frame count of a song, the relationship between o and c can be expressed as o = Wc, where W is the window matrix expansion from c to o. The optimal static feature vector sequence is as follows:

$$\hat{c} = \arg\max_{c} P(o \mid \lambda) = \arg\max_{c} N(w_c \mid \mu, \Sigma) \tag{22}$$

λ is the parameter set, and $N(\cdot \mid \mu, \Sigma)$ denotes the Gaussian distribution with mean vector μ and covariance matrix Σ. μ is the output of the trained neural network. The optimal value $\hat{c}$ of the static feature sequence is:

$$\hat{c} = PW^T\Sigma^{-1}\mu, \quad P = (W^T\Sigma^{-1}W)^{-1} \tag{23}$$

Training the DNN aims to maximize the likelihood function L, defined as follows:

$$L = P(o \mid \lambda) = N(o \mid \mu, \Sigma) = \prod_{t=1}^{T} N(o_t \mid \mu_t, \Sigma_t) \tag{24}$$

In traditional DNN-based vocal synthesis, equation (24) employs a frame-level objective function to train the DNN, while equation (22) uses a sequence-level objective function to generate parameters. To resolve the level inconsistency between training and synthesis, trajectory training is incorporated into the training phase of DNN-based vocal synthesis systems. The traditional likelihood function is replaced with a trajectory likelihood function $o = Wc$, which strengthens the relationship between static and dynamic features. The trajectory likelihood function for the static feature vector is:

$$L_{Trj} = \frac{1}{z} P(o \mid \lambda) = P(c \mid \lambda) = N(c \mid \bar{c}, P) \tag{25}$$

Here, Z denotes normalization, and the mean vector c is identical to the static feature sequence generated by equation (23). The parameter sequence λ can be estimated by maximizing the trajectory likelihood function.

3.2.3 Tremolo Model

Singing encompasses both lyrics and melody, so compared to speech synthesis, vocal synthesis must also account for singing conventions. One such convention is vibrato. To make the synthesized vocals sound more natural and closer to real singing, a vibrato module has been incorporated into the model.

Here, vibrato is modeled using a sinusoidal curve with periodic fluctuations in fundamental frequency. For the i-th vibrato segment $[t_i^{(s)}, t_i^{(e)}]$, the vibrato at frame $v(\cdot)$ can be expressed as:

$$v(m_a(t), m_f(t), i) = m_a(t)\sin(2\pi m_f(t) f_s(t - t_i^{(s)})) \tag{26}$$

where $m_a(t), m_f(t)$ and f_s represent the amplitude of the fundamental frequency of the vibrato, the frequency of the fundamental frequency of the vibrato, and the frame shift, respectively. $m_a(t)$ and $m_f(t)$ are both present in the acoustic feature vector.

3.2.4 Delay Model

Beyond the vibrato model, another crucial aspect of choral vocal synthesis is that the onset time of the actual singing should precede the corresponding note's start time.

Theoretically, chorus singing determines start times, pitch, and other information based on the musical score. The beat and phoneme duration of the singing must also follow the score. However, strictly adhering to the score can make the alignment between the chorus singing and the musical notes appear stiff. Therefore, in actual chorus singing, the onset time is always slightly earlier than the corresponding note in the score. To ensure the synthesized singing sounds closer to real singing, a delay model is incorporated. In the HMM-based choral singing synthesis system, each note's timing is modeled using a Gaussian distribution. HSMM is used to locate the start time of each note in the vocal database. Subsequently, a delay model is trained, and decision tree-based context clustering is applied to this delay model. Here, a DNN replaces HMM for training the delay model. The duration of each phoneme, the time interval between the onset of a note and the vocal onset, and the duration of the phoneme corresponding to each note are also modeled using a DNN. The duration of the Kth phoneme in the nth note is defined as follows:

$$d_{nk}^{(DNN)} = L_n \cdot \mu_{nk} / \sum_{k=1}^{k_n} \mu_{nk} \tag{27}$$

In the above equation, L_n denotes the length of the nth note after applying delay, k_n represents the number of phonemes contained within the nth note, and μ_{nk} is the output value of the DNN-based duration model corresponding to the kth phoneme among the n notes.

During synthesis, first determine the note durations from the target musical score. Based on the delay model, establish the delay time for each note and define its boundaries. Combine the duration model with the note lengths to further predict the durations of each minimal primitive.

4 Experimental Results and Analysis

4.1 Choral Vocal Separation Evaluation

4.1.1 Choral Sound Source

Collect 150 choral songs to train the choral voice separation model, and use the remaining 20 choral songs to test the algorithm's separation effectiveness.

The specific training process for the choral voice separation model based on deep neural networks is as follows: First, extract choral segments and accompaniment audio segments from the audio library. For the accompaniment audio segments, slicing processing is also required. Next, audio segments of equal length (Chorus) and accompaniment audio segments (Accompany) were combined to form the choral song (Union). Feature extraction was then performed on the choral song, choral segments, and accompaniment audio separately. Feature extraction included steps such as frame segmentation, short-time Fourier transform, and extraction of the log-power spectrum. The model training process then commences, with inputs comprising the log-power spectral features of the Union, Chorus, and Accompany. Upon completion of model training, a deep neural network-based vocal separation model is obtained. This model is subsequently applied to vocal separation tasks.

4.1.2 Evaluation Indicators

In practical computational applications, the PESQ algorithm is currently the mainstream audio quality assessment method. The specific implementation process of the PESQ objective audio quality evaluation algorithm is as follows: First, the original audio and separated audio signals

are level-aligned to a standard monitoring level. Input filters are applied to both audio signals. The signals are arranged chronologically and then processed through an auditory transformation similar to PSQM. This transformation also involves linear filtering and gain-varying equalization within the system. Two distortion parameters are extracted from the perturbation (the difference between the transformed signals), aggregated across frequency and time, and mapped to a predicted Subjective Mean Opinion Score (MOS).

4.1.3 Experimental Results

This paper employs audio samples combining choral vocals with instrumental accompaniment to train a deep neural network-based choral vocal separation model (DRNN). The trained model was then applied to separate the sound sources, yielding WAV-format audio files suitable for human subjective listening evaluation. The separation results were assessed using the Perceptual Evaluation of Speech Quality (PESQ) scale. Figure 3 illustrates the PESQ ratings for the three audio samples after model separation. Since the choir vocals and accompaniment were mixed at seven distinct signal-to-noise ratios (SNR), the results display the PESQ scores for each audio type across these seven SNR levels. The original mixed audio achieved an average PESQ score of 2.67 across all seven SNRs. The separated choir vocals and accompaniment achieved average PESQ scores of 3.32 and 1.32, respectively.

As the SNR progressively increased from -4 to 20, the PESQ ratings also rose significantly. Both the original vocal track and the separated vocal audio achieved their highest PESQ scores at an SNR of 20 dB, registering 3.31 and 3.76 respectively. The separated instrumental audio maintained a relatively stable PESQ score between 1.30 and 1.36, peaking at 1.36 at 0 dB. This indicates that as the signal-to-noise ratio increases from -4 to 20, the proportion of vocal content within the song audio grows progressively larger. Consequently, the separated vocal track possesses greater energy, leading to improved evaluation results. However, increasing the signal-to-noise ratio has little effect on the quality of the separated instrumental track.

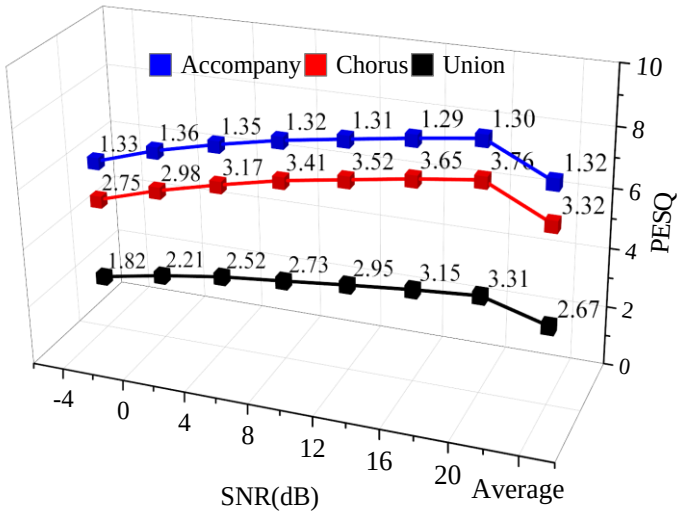

Figure 3: PESQ evaluation of three audio frequencies

To test the impact of iteration counts on separation results during model training, the speech source separation model was trained using iteration counts of 50, 60, and 70. The resulting PESQ scores for the separated choir audio under various signal-to-noise ratios are shown in Figure 4. For training iterations of 50, 60, and 70, the average PESQ ratings of the separated audio at different SNRs were 3.26, 3.28, and 3.27, respectively. It can be observed that the PESQ ratings across the three iteration counts are broadly comparable. For a single iteration

count, the PESQ scores of the speech audio continuously increase. However, at the same SNR, the PESQ score for 50 iterations is significantly lower than that for 60 iterations. Moreover, the PESQ score for 60 iterations is slightly higher than that for 70 iterations. This indicates that the model achieves optimal performance at 60 iterations. Therefore, considering time constraints, selecting 60 iterations is appropriate for the deep neural network-based choral voice separation model.

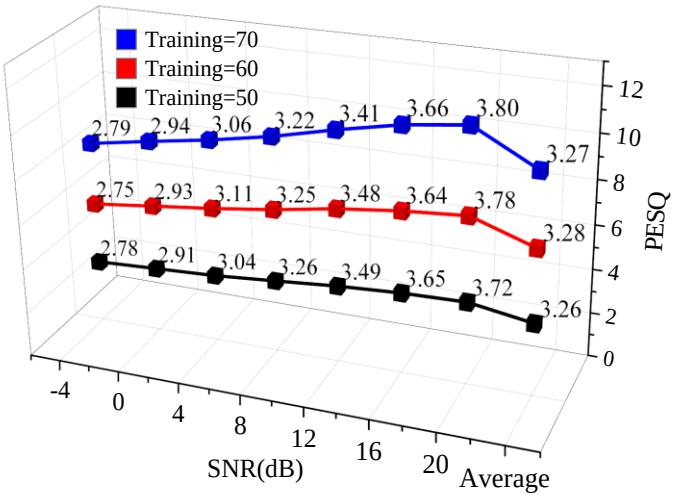

Figure 4: PESQ evaluation of the model under different training times

To analyze the performance of the deep neural network-based choral sound separation model, the L-MMSE algorithm, BP algorithm, DNN algorithm, and the DRNN choral sound separation model proposed in this paper were selected for comparative evaluation. Figure 5 presents the PESQ comparison results for the four models. The average PESQ score for the choral sound obtained after source separation using the deep neural network-based choral sound separation model employed in this study was 3.31. In contrast, the average PESQ scores for the L-MMSE algorithm, BP algorithm, and DNN algorithm were 2.61, 2.45, and 2.48, respectively. This indicates that the deep neural network-based choral sound separation model adopted in this study demonstrates superior performance in choral sound separation.

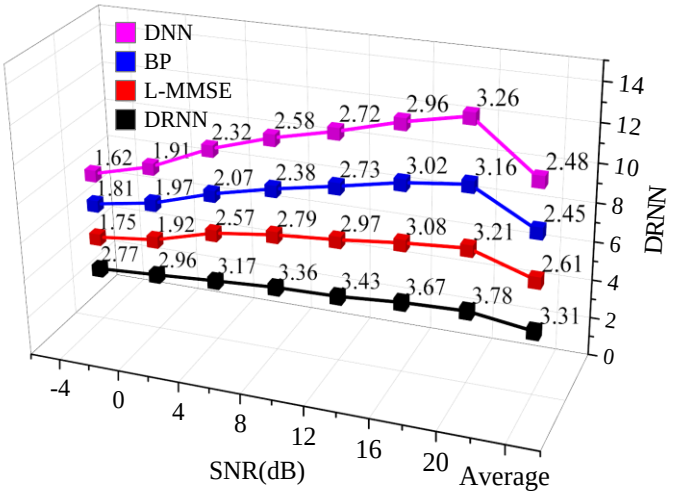

Figure 5: PESQ evaluation comparison of the four models

4.2 Evaluation of Choral Voice Synthesis

4.2.1 Dataset

The dataset selected for this paper is the JS Fake Chorales dataset, which is an extension of the JS Bach Chorales dataset. It is derived from Bach's chorales, expanded with similar music, and includes manually annotated chord progressions. The dataset comprises 400 four-part choral pieces.

4.2.2 Evaluation Indicators

There is no consistent evaluation standard for measuring music generation samples. From a model generation perspective, the Token Error Rate (TER) is selected as an indicator to measure the model's modeling capability. From a musicality perspective, statistical metrics for chord tones and non-chord tones provide a measure of harmonic coherence. This paper selects three chord/melody harmonic metrics as methods for measuring the musicality of choral music: Chord-to-Nonchord Tone Ratio (CTnCTR), Pitch Consistency Score (PCS), and Melody-Chord Tone Distance (MCTD). Additionally, the RS indicator is selected as a quantitative evaluation of synthesized music by professional musicians, normalized to a range of 0.2 to 1.

4.2.3 Musicality Metrics

This paper employs the metrics from the previous section to measure musicality, comparing them against Bach's original work metrics. The musicality measurement results are shown in Figure 6, where h represents the gamma sampling coefficient. When compared to Bach's original work, the results demonstrate that the proposed choral voice synthesis model has achieved near-identical metrics to Bach's work under the current evaluation criteria (within a 10% error margin). The error ranges for the four musicality metrics—RS, MCTD, PCS, and CTnCTR—are 0.92%–7.20%, 2.59%–8.54%, 1.63%–8.65%, and 1.40%–9.42%, respectively, demonstrating the superiority of the DNN-based choral voice synthesis model. In the task of providing specified harmonies for a given melody, the model proposed in this paper has already approached the level of human composers.

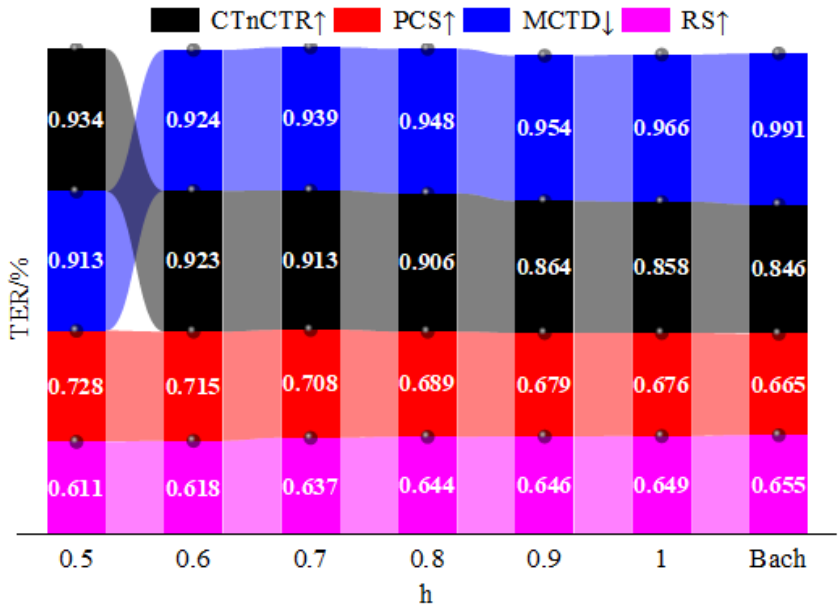

Figure 6: The results of the musical metrics indicators

4.2.4 Visual Analysis

Visualizing synthesized music through a piano roll provides an intuitive representation of the generated melody lines. The piano roll in Figure 7 depicts a four-part musical melody. A piano roll can be conceptualized as a two-dimensional grid, where the x-axis represents the duration of each time step and the y-axis represents pitch. The annotated (x, y) points indicate that the y-pitch is activated at time x.

This paper analyzes the generated samples. Comparing music generated by the original Transformer structure's self-attention mechanism and the DNN-based choir vocal synthesis model, we perform Pianoroll visualization analysis on the generated choir music. The top figure shows music generated using the choir vocal synthesis model relative to DNN, while the bottom figure displays music generated using the Transformer attention mechanism, presented via Pianoroll with the horizontal axis representing the number of notes. As highlighted in the figure, after generating 400 notes, the DNN-based model produces a longer effective length, whereas the Transformer model generates prolonged monophonic sequences that diminish musicality. This demonstrates that our model generates more musically coherent sequences while avoiding the production of redundant notes.

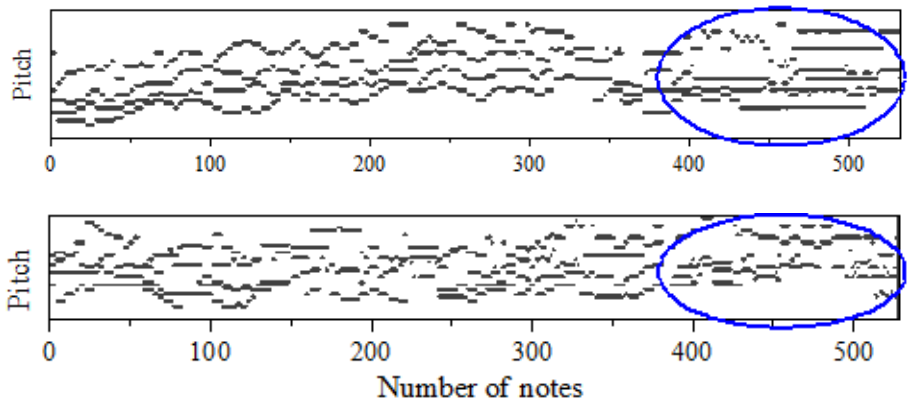

Figure 7: Visual analysis of synthetic music

4.2.5 Comparative Testing

This paper selected DeepChoir, TonieNet, and DeepBach models for comparative experiments. The comparison of different music synthesis models is shown in Figure 8. The Token Error Rate was used as the metric to evaluate modeling capability. Different models were tested on the same validation set, and the error rate (%) of incorrect notes in the output was calculated. Since the input format for RNN-based models differs from the proposed model—where the soprano part is used as input—the soprano output of corresponding models was set to zero and excluded from generation comparisons.

Experiments demonstrate that the proposed DNN model achieves superior modeling capability compared to other models. Its TER metrics for the four distinct voice parts range from 4.25% to 4.66%, all below 5%, whereas the TER metrics for other music synthesis models exceed 5% across the board. Comparative testing reveals that the model exhibits higher prediction error rates for the soprano part, with a TER of 4.66%. During inference, predicting the soprano's main melody proves more challenging than generating the harmonies accompanying the main melody.

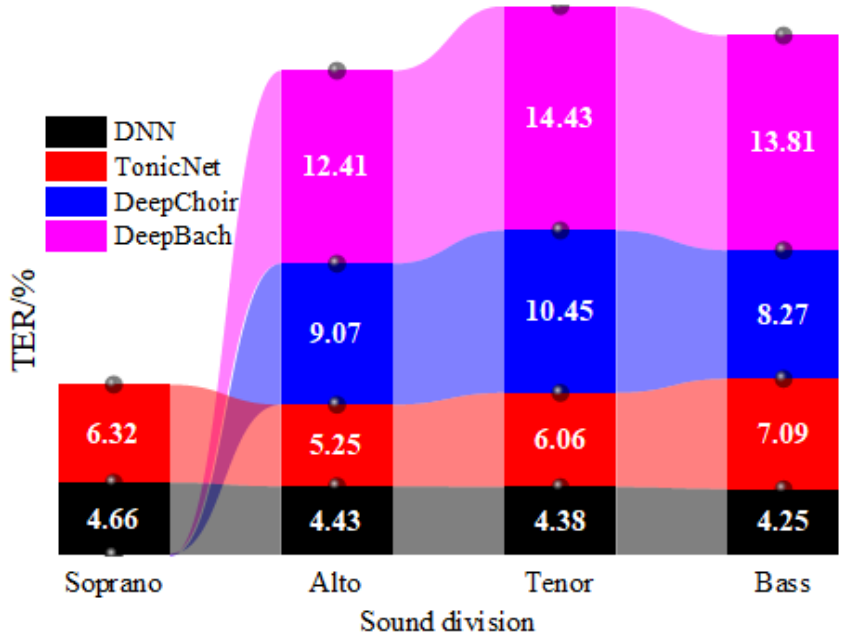

Figure 8: Comparison of Token error rate(TER)

5 Conclusion

This paper investigates audio separation and digital music synthesis by constructing a vocal separation model and a backing vocal model using deep recurrent neural networks (DRNN) and DNN neural networks, respectively. The performance of both models in separating vocals and synthesizing backing vocals was evaluated using a collected dataset of choral songs. Key findings are as follows:

(1) Under varying signal-to-noise ratios, the DRNN model achieved a mean PESQ score of 3.32 for separated vocal tracks, significantly outperforming the original audio (2.67) and separated instrumental tracks (1.32). Furthermore, the PESQ scores for the separated vocal tracks from this model improved by 26.82%, 35.10%, and 33.47% compared to the L-MMSE algorithm, BP algorithm, and DNN algorithm, respectively. The DRNN model demonstrated superior performance in vocal separation.

(2) The music generated by the DNN choir voice synthesis model exhibits error values below 10% compared to the original across four musicality metrics. It also produces longer monophonic sequences, enhancing musicality. Regarding model performance, the TER metric for music generated by the DNN model remains below 5% across different vocal parts, whereas the TER metric for the comparison model ranges from 5.25% to 14.43%. Experiments demonstrate that the proposed model outperforms existing choir voice generation models.

The application of deep learning intelligence in music creation represents a profound revolution, breaking numerous constraints of traditional composition and empowering creators with greater possibilities. From transforming creative tools to revolutionizing dissemination methods, deep learning intelligence infuses music with renewed vitality. However, while embracing modern technology, we must not overlook the intrinsic essence of musical artistry. We should actively explore pathways for integrating technology and artistry, propelling music creation toward new heights.

References

[1] Crocco, M., Cristani, M., Trucco, A., & Murino, V. (2016). Audio surveillance: A systematic review. ACM Computing Surveys (CSUR), 48(4), 1-46.

[2] Duong, N. Q., & Duong, H. T. (2015). A review of audio features and statistical models exploited for voice pattern design. arXiv preprint arXiv:1502.06811.

[3] Mistry, Y. D., Birajdar, G. K., & Khodke, A. M. (2023). Time-frequency visual representation and texture features for audio applications: a comprehensive review, recent trends, and challenges. Multimedia Tools and Applications, 82(23), 36143-36177.

[4] Purwins, H., Li, B., Virtanen, T., Schlüter, J., Chang, S. Y., & Sainath, T. (2019). Deep learning for audio signal processing. IEEE Journal of Selected Topics in Signal Processing, 13(2), 206-219.

[5] Richard, G., Smaragdis, P., Gannot, S., Naylor, P. A., Makino, S., Kellermann, W., & Sugiyama, A. (2023). Audio signal processing in the 21st century: The important outcomes of the past 25 years. IEEE Signal Processing Magazine, 40(5), 12-26.

[6] Ling, Z. H., Kang, S. Y., Zen, H., Senior, A., Schuster, M., Qian, X. J., ... & Deng, L. (2015). Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and future trends. IEEE Signal Processing Magazine, 32(3), 35-52.

[7] Golovko, V. A. (2017). Deep learning: an overview and main paradigms. Optical memory and neural networks, 26(1), 1-17.

[8] Alsajri, A. K. S., & Hacimahmud, A. V. (2023). Review of deep learning: Convolutional neural network algorithm. Babylonian Journal of Machine Learning, 2023, 19-25.

[9] Zhang, Z., Geiger, J., Pohjalainen, J., Mousa, A. E. D., Jin, W., & Schuller, B. (2018). Deep learning for environmentally robust speech recognition: An overview of recent developments. ACM Transactions on Intelligent Systems and Technology (TIST), 9(5), 1-28.

[10] Ning, Y., He, S., Wu, Z., Xing, C., & Zhang, L. J. (2019). A review of deep learning based speech synthesis. Applied Sciences, 9(19), 4050.

[11] Shihab, A. I. (2023). Voice separation and recognition using machine learning and deep learning a review paper. Journal of Al-Qadisiyah for computer science and mathematics, 15(3), Page-11.

[12] Ozerov, A., Philippe, P., Bimbot, F., & Gribonval, R. (2007). Adaptation of Bayesian models for single-channel source separation and its application to voice/music separation in popular songs. IEEE Transactions on Audio, Speech, and Language Processing, 15(5), 1564-1578.

[13] Mysore, G. J., Smaragdis, P., & Raj, B. (2010, September). Non-negative hidden Markov modeling of audio with application to source separation. In International Conference on Latent Variable Analysis and Signal Separation (pp. 140-148). Berlin, Heidelberg:

Springer Berlin Heidelberg.

[14] Tzinis, E., Wisdom, S., Hershey, J. R., Jansen, A., & Ellis, D. P. (2020, May). Improving universal sound separation using sound classification. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 96-100). IEEE.

[15] Huang, P. S., Kim, M., Hasegawa-Johnson, M., & Smaragdis, P. (2014, May). Deep learning for monaural speech separation. In 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1562-1566). IEEE.

[16] Chen, J., Liu, C., Xie, J., An, J., & Huang, N. (2022). Time–frequency mask-aware bidirectional lstm: A deep learning approach for underwater acoustic signal separation. Sensors, 22(15), 5598.

[17] Poh, Y. Y., Grooby, E., Tan, K., Zhou, L., King, A., Ramanathan, A., ... & Marzbanrad, F. (2024). NeoSSNet: Real-time neonatal chest sound separation using deep learning. IEEE Open Journal of Engineering in Medicine and Biology, 5, 345-352.

[18] Hayes, B., Shier, J., Fazekas, G., McPherson, A., & Saitis, C. (2024). A review of differentiable digital signal processing for music and speech synthesis. Frontiers in Signal Processing, 3, 1284100.

[19] Lu, P., Wu, J., Luan, J., Tan, X., & Zhou, L. (2020). Xiaoicesing: A high-quality and integrated singing voice synthesis system. arXiv preprint arXiv:2006.06261.

[20] Saleem, N., Gao, J., Irfan, M., Verdu, E., & Fuente, J. P. (2022). E2E-V2SResNet: Deep residual convolutional neural networks for end-to-end video driven speech synthesis. Image and Vision Computing, 119, 104389.

[21] Nishimura, M., Hashimoto, K., Oura, K., Nankaku, Y., & Tokuda, K. (2016, September). Singing Voice Synthesis Based on Deep Neural Networks. In Interspeech (pp. 2478-2482).

[22] Nakamura, K., Takaki, S., Hashimoto, K., Oura, K., Nankaku, Y., & Tokuda, K. (2020, May). Fast and high-quality singing voice synthesis system based on convolutional neural networks. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 7239-7243). IEEE.

[23] Guan, Y., Wei, Q., & Chen, G. (2019). Deep learning based personalized recommendation with multi-view information integration. Decision Support Systems, 118, 58-69.

[24] Ain, Q. T., Ali, M., Riaz, A., Noureen, A., Kamran, M., Hayat, B., & Rehman, A. U. (2017). Sentiment analysis using deep learning techniques: a review. International Journal of Advanced Computer Science and Applications, 8(6).