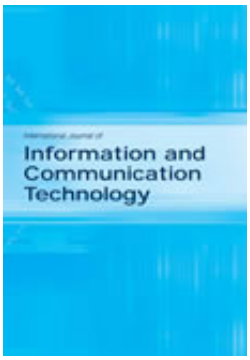


International Journal of Information and Communication Technology

ISSN online: 1741-8070 - ISSN print: 1466-6642
https://www.inderscience.com/ijict

Sparse coding-based vocal music feature extraction and real-time transmission

Fangzi Zhang, Jinyi Hu

DOI: 10.1504/IJICT.2025.10074749

Article History:

Received:	30 December 2024
Last revised:	13 January 2025
Accepted:	14 January 2025
Published online:	01 December 2025

Sparse coding-based vocal music feature extraction and real-time transmission

Fangzi Zhang

School of Music,
Hunan International Economics University,
Changsha, 410000, China
Email: 2021t0991@pwu.edu.ph

Jinyi Hu*

School of Humanities and Music,
Hunan Vocational College of Science and Technology,
Changsha, 410000, China
Email: hujinyi1104@163.com
*Corresponding author

Abstract: Traditional audio compression and transmission methods struggle with bandwidth usage and transmission delay, thereby creating a growing need for a real-time audio transmission. This work presents a sparse coding-based approach for vocal audio feature extraction and real-time transmission (SCTRT) to handle these difficulties. By means of sparse coding approaches, the model efficiently compresses and extracts audio information, hence lowering data redundancy and improving transmission efficiency. Three components make up the model: real-time transmission and recovery, feature extraction and compression, and audio capture and pre-processing, guaranteeing low latency and effective transmission of audio signals. In terms of compression ratio, audio quality and transmission delay, the experimental findings reveal that the SCTRT model is particularly appropriate for real-time audio transmission applications since it has notable benefits over conventional techniques.

Keywords: sparse coding; vocal feature extraction; audio compression; real-time transmission.

Reference to this paper should be made as follows: Zhang, F. and Hu, J. (2025) 'Sparse coding-based vocal music feature extraction and real-time transmission', *Int. J. Information and Communication Technology*, Vol. 26, No. 42, pp.1–17.

Biographical notes: Fangzi Zhang received her Master's degree from the Ukraine National Normal University in 2009. She is currently a Lecturer in Hunan International Economics University. Her research interests include machine learning, and vocal music teaching.

Jinyi Hu received a Doctorate in Musicology from the Philippine Women's University in 2024. He is currently working in Hunan Vocational College of Science and Technology. His research interests include machine learning, vocal music teaching and music signal processing.

1 Introduction

In various disciplines like voice communication, online education, telemedicine, virtual reality, online music performance, and so on, audio signal processing and real-time transmission technology become even more crucial with the ongoing development of information technology (Latif et al., 2020; Gupta et al., 2019). In the realm of audio communication and processing, solving the fundamental problem of lowering transmission delay and bandwidth consumption while guaranteeing audio quality in real-time audio transmission becomes critical. Although conventional audio transmission technologies – especially those based on compression and coding techniques – such as MP3, AAC, etc. – can efficiently compress audio data – in the face of the demand for high-quality audio transmission and real-time processing their compression effect and processing efficiency face a certain bottleneck (Cunningham and McGregor, 2019). Particularly in complicated surroundings, elements like noise, signal attenuation, and bandwidth constraints aggravate the difficulties with real-time audio transmission.

Emerging signal processing technique sparse coding technology offers great promise in audio signal processing and compression in recent years (Umapathy et al., 2010). Sparse coding uses the sparsity of the signal for representation to effectively compress and reconstruct the signal by greatly lowering the signal's redundancy without sacrificing any substantial information. Unlike conventional audio compression techniques, sparse coding not only concentrates on the compression impact of audio signals but also allows a more compact and accurate signal representation by thus efficiently extracting the sparse aspects of the signal (Valenzise et al., 2009). Although considerable progress has been made in the field of audio signal processing thanks to sparse coding, how to combine it with real-time audio transmission technology – especially when real-time transmission must be quick and have low latency (Wu et al., 2001).

Most current studies on audio feature extraction and compression follow transform-based coding, learning-based feature extraction, and deep learning model application. Deep learning technology is rapidly developing, thus convolutional neural networks (CNN) and recurrent neural networks (RNN), etc. are extensively applied in audio feature extraction (Deng et al., 2020; Singh, 2024), which is able to automatically learn and extract high-level features in audio signals, so providing a stronger representation capability than conventional methods. These deep learning techniques still need additional optimisation though they have issues including large computing overhead and long response times in real-time transmission circumstances.

While some circumstances have seen improvement in current techniques for extracting audio features and compressing them, they still have issues with latency, bandwidth utilisation, and computing complexity when transmitting high-quality audio in real-time (Michelsanti et al., 2021). This work presents a sparse coding-based approach for voice audio feature extraction and real-time transmission, commonly referred to as SCTRT, in order to address these challenges.

This work combines the following novelties:

1 Low-latency and efficient audio compression and transmission model: To drastically lower the data volume while maintaining the audio quality, the SCTRT model presented in this work combines effective audio compression techniques with sparse coding. In environments with high real-time needs, the SCTRT model not only achieves notable compression ratio and audio quality but also lowers the

transmission delay and improves the transmission efficiency when compared with conventional audio compression technologies (e.g., MP3 or AAC).

2 Combination of sparse coding and real-time audio transmission: The SCTRT model solves bandwidth consumption and transmission latency of audio signals in real-time transmission by merging sparse coding technology with real-time audio transmission, therefore transcending the constraints of conventional audio processing techniques. While signal compression lets the audio signals be effectively sent with limited bandwidth and quick transmission time, sparse coding preserves the high-quality expression of audio qualities.

3 Multi-module co-optimisation architecture: Using three interdependent modules – audio capture and pre-processing, feature extraction and compression, and real-time transmission and recovery – the SCTRT model while the transmission module combines low-latency compression techniques to preserve effective performance in many contexts, the feature extraction and compression module optimises the extraction of audio features by introducing sparse coding.

These innovations not only enrich the theoretical research in the field of audio processing and transmission, but also provide an efficient, low-latency and low-bandwidth audio transmission solution for practical applications, especially in real-time audio transmission, voice communication and audio streaming scenarios.

2 Relevant technologies

2.1 Vocal audio feature extraction techniques

A fundamental stage in audio processing, vocal audio feature extraction is also necessary for further analysis, classification, and transmission activities. Usually, conventional feature extraction techniques extract representative features from the frequency and time domain information of the audio stream. One of the most often used techniques is the Mel frequency cepstral coefficients (MFCC) (Pawar and Kokate, 2021), which, by replicating the auditory characteristics of the human ear, transforms audio data into frequency coefficients at the Mel frequency scale, fit for characterising traits as pitch and timbre, see Figure 1.

In particular, the audio signal $S(t, f)$ is initially obtained by a short-time Fourier transform (STFT): time-frequency spectrum:

$$S(t, f) = \left| \int_{\infty}^{\infty} x(\tau) w(t-\tau) e^{-j2\pi ft} d\tau \right|^2 \tag{1}$$

where t is the time; $w(t-\tau)$ is the window function; f is the frequency. The Meier filter bank then completes the frequency scale transformation of the spectrum to produce the Meier energy spectrum E_m:

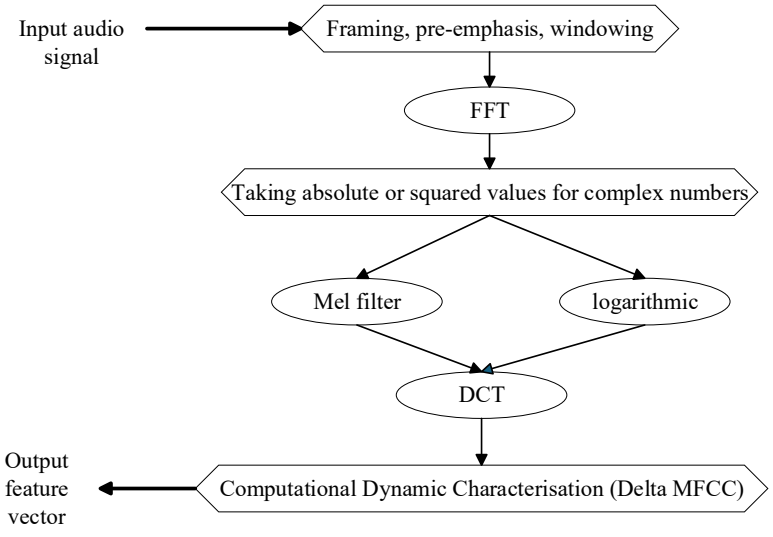
$$E_m = \sum_f \left| S(t, f) \right|^2 \cdot H_m(f) \tag{2}$$

$H_m(f)$ models the Mayer filter response function. Finally, we derive the MFCC coefficients C_m by use of the discrete cosine transform (DCT) and logarithmic processing:

$$C_m = DCT\left(\log\left(E_m\right)\right) \tag{3}$$

The basic frequency characteristics of the audio signal can be expressed with these MFCC coefficients.

Figure 1 MFCC principle



Spectrogram is another widely used method of feature extraction since it divides the audio signal into many short-time windows and computes the spectrum of every window (Nasr et al., 2018), therefore obtaining a two-dimensional representation of the audio signal in the time-frequency plane. Especially in uses where the variation of the audio signal needs to be accurately analysed, the spectrogram is appropriate for capturing the periodic and non-periodic components of the signal since it shows the energy distribution of the audio signal at various times and frequencies. The formula determines the spectrogram:

$$S(t, f) = \left| \sum_{\tau=-\infty}^{\infty} x(\tau) w(t-\tau) e^{-j2\pi f t} \right| \tag{4}$$

Unlike MFCC, spectrograms often preserve more original frequency information but often contain redundancy as a result and cannot efficiently shrink the information dimension.

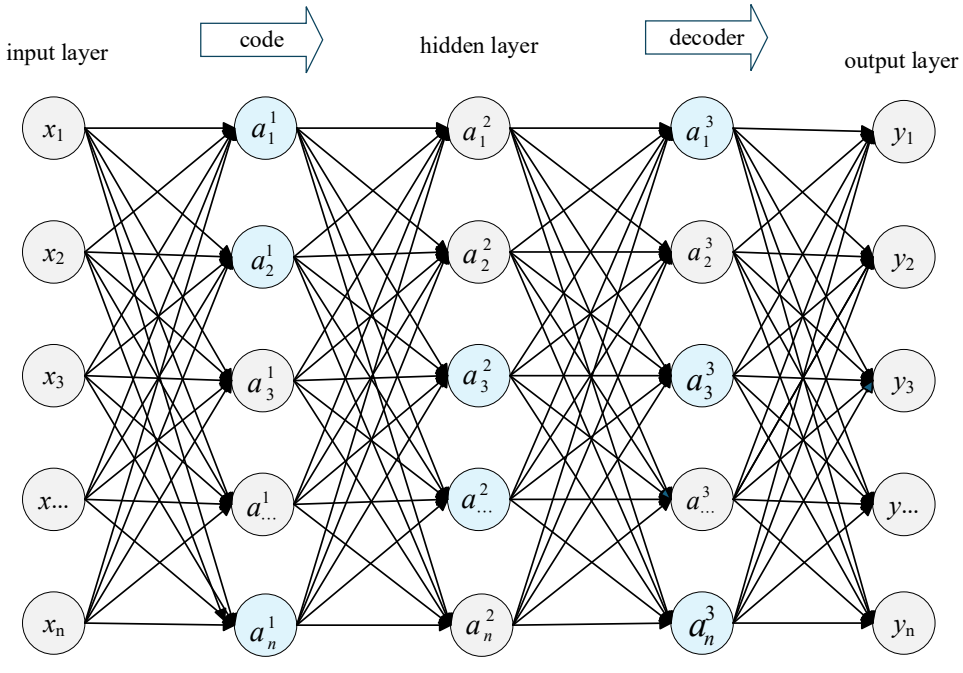
Nevertheless, conventional feature extraction techniques only capture the most fundamental aspects of audio sources. Usually unable to delve deeply into the more intricate components, they may run across problems with duplicate information when handling sophisticated audio signals.

Consequently, in this work we will combine sparse coding methods to maximise the vocal audio feature extraction. The sparse coding method eliminates needless data and improves the signal representation by use of a sparse dictionary and a straight line with few dictionary items for encoding the sound signal. Sparse coding is more compact and economical in feature representation than conventional techniques, which also help to more precisely extract the essential elements of audio signals.

2.2 Sparse coding techniques

Especially for high-dimensional, redundant, and complex data, sparse coding methods have grown to be a valuable instrument in signal processing and feature extraction (Qayyum et al., 2019). Whereas conventional feature extraction techniques usually depend on manually creating features, sparse coding is able to automatically learn a dictionary from the data and represent the signal as a linear combination of a limited number of items in that dictionary. In the field of vocal audio signal processing, this sparse representation has been extensively applied as it may efficiently eliminate duplicate information and increase the efficiency of signal processing (Xu et al., 2022); see Figure 2.

Figure 2 Sparse coding principle (see online version for colours)



Sparse coding is essentially based on the use of a dictionary to represent the signal $x \in \mathbb{R}^m$ and ensure that the representation is as sparse as feasible, hence limiting the number of non-zero coefficients. Usually, the optimisation aim of sparse coding is expressed by the following equation:

$$J(D, \alpha) = \|x - D\alpha\|_2^2 + \lambda \|\alpha\|_1 \tag{5}$$

Each column $d_i \in \mathbb{R}^m$ denotes a basis vector in the dictionary, denoted as $D = [d_1, d_2, \ldots, d_k]$, whereas $\alpha \in \mathbb{R}^k$ is the sparse coefficient vector, therefore indicating the sparse representation of the signal x in the dictionary D. The l_1 paradigm $||\alpha||_1$ allows one to manage the sparsity of the sparse coefficients α by enabling the signal x to be approximated by a linear combination of a limited number of basis vectors in the dictionary D.

In sparse coding, the reconstruction error is represented as E by the difference between the signal x and the signal rebuilt by the dictionary D:

$$E = \|x - D\alpha\|_2^2 \tag{6}$$

Minimising the reconstruction error E will help us to get a better dictionary D and sparsity coefficient α by letting a limited number of dictionary bases to reflect the signal x.

Orthogonal matching pursuit (OMP) is one of the often used techniques to attain sparse coding (Yang and De Hoog, 2015). Selecting the basis vectors in the dictionary D that best fit the target signal, the OMP method gradually approximates the signal step-by-step. We update the sparse coefficients every time we choose the dictionary base best lowering the reconstruction error. Assuming E_k as the current reconstruction error, the dictionary basis d_i is chosen at each iteration to minimise the reconstruction error; the update formula can be stated as follows:

$$E_{k+1} = \left\| x - \sum_{i=1}^{k+1} d_i \alpha_i \right\|_2^2 \tag{7}$$

where $\sum_{i=1}^{k+1} d_i \alpha_i$ is the fresh sparse representation and E_{k+1} represents the reconstruction error following the $k + 1$st iteration.

Moreover important is the dictionary update mechanism for sparse coding (Cui and Prasad, 2016). With a set of signal samples, for every signal x_i we have the appropriate sparsity coefficient α_i. Dictionary updating aims to minimise the reconstruction error and the sparsity constraints for every sample thereby optimising the dictionary D. The following formula especially captures the intention of dictionary updating:

$$D = \arg\min_D \sum_{i=1}^{N} \left(\|x_i - D\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1 \right) \tag{8}$$

By maintaining the dictionary and sparse coefficients current all the time, sparse coding can learn a vocabulary that can faithfully depict the signal and execute accurate reconstruction with sparse coefficients.

We can introduce the dictionary learning approach, which iteratively learns the dictionary and sparse coefficients, so optimising the representation and thereby enhancing the accuracy of feature extraction. Dictionary learning aims to update the dictionary and sparse coefficients by means of an optimisation problem following which.

$$D^{new} = \arg\min_{D} \sum_{i=1}^{N} \left(\|x_i - D\alpha_i\|_2^2 \right) \tag{9}$$

Using the following equation helps to decrease the reconstruction error while updating the sparsity coefficient α:

$$\alpha_i = \arg\min_{\alpha_i} \left(\|x_i - D\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1 \right) \tag{10}$$

By means of the above-mentioned dictionary learning method, we can progressively improve the dictionary to make it more fit for the present dataset, so enhancing the representation of the audio signal and the feature extraction accuracy.

2.3 Audio compression and real-time transmission technology

Particularly in cases of limited bandwidth and low latency, audio compression and real-time transmission techniques are rather crucial in voice audio feature extraction systems. Real-time transmission methods paired with a good audio compression technique can efficiently lower the data load and guarantee high-quality audio signal transmission, thereby helping to alleviate the transmission bottleneck of audio signals. Combining audio compression with sparse coding not only lowers duplicate information but also preserves signal integrity across transmission (Nassra and Capella, 2023).

Audio signal compression's core goal is to minimise reconstruction mistakes by representing audio data with less bits. In approaches based on sparse coding, the audio signal is expressed as a linear combination of dictionary basis vectors. Usually, the optimisation issue is stated as a maintenance of a good quality of the compressed signal as:

$$\min_{\alpha} \|x - D\alpha\|_2^2 \text{ subject to } \|\alpha\|_0 \le T \tag{11}$$

where T is a given sparsity threshold and $||\alpha||_0$ stands for the sparsity – that is, the count of non-zero coefficients. To reconstruct the signal for compressing needs, the compression process chooses as few basis vectors as feasible.

Practically, audio signal compression calls for not just reduced data volume but also great fidelity (Lie and Chang, 2006). By progressively choosing the most relevant basis vectors from the dictionary, the matching pursuit or OMP algorithm solves the sparse coefficient. Usually, the update step of the OMP algorithm is described by the following equation:

$$a_k = \arg\min_{\alpha} \left(\|x - D\alpha\|_2^2 \right) \text{ subject to } \|\alpha\|_0 = k \tag{12}$$

where k stands for the count of chosen basis vectors. OMP may efficiently achieve effective sparse coding and drastically lower the computational complexity by iterative optimisation.

Apart from compression, low latency and great efficiency in real-time transmission have to be given due consideration. Often lowering transmission latency to guarantee effective audio signal distribution over a network are the real-time transport protocol (RTP) and the user datagram protocol (UDP). Forward error correction (FEC) methods

are utilised to make the system more dependable even if UDP lacks a means to ensure that packets are delivered correctly and so the delay resulting from packet acknowledgement can be avoided while transferring audio over UDP. By including redundancy information into the packet, FEC lets the receiver recover data should data loss occur.

By adding redundant bits to every packet, FEC guarantees that the receiver may recover the signal even if some data is lost assuming that audio signals are sent in packets of size N. Although it adds more data overall, redundant information helps the transmission be more robust. Usually, the equation below shows the maximisation of the recovery capacity:

$$P_{recovery} = 1 - (1 - p)^k \tag{13}$$

where $P_{recovery}$ is the possibility of recovery; p denotes the possibility of packet loss; k denotes the number of extra messages. The FEC can provide a high recovery rate in transmission by suitable design of the redundancy, therefore improving the system performance in low-quality network conditions.

The data compression and transmission procedure must balance computational complexity, bandwidth usage, and delay in the real-time audio signal transmission. With each data packet having a size of N_{packet} and a transmission delay of T_{delay}, we may show the total transmission delay, T_{total}, assuming that we must broadcast the audio signal over the network following compression:

$$T_{total} = T_{encode} + T_{packet} + T_{decode} \tag{14}$$

where T_{encode} is the encoding delay; T_{packet} is the packet transmitting delay; T_{decode} is the decoding time. Usually, optimising the encoding and decoding techniques helps to avoid delays so guaranteeing real-time performance.

Combining sparse coding and FEC approaches solves the problem of audio signals in restricted bandwidth and low-latency transmission essentially. While sparse coding not only lowers redundant information in audio compression but also guarantees high-quality reconstruction of audio signals; in transmission, the UDP protocol and the FEC technique improve the robustness and dependability of transmission; at the same time, the use of variable-length frames and adaptive framing strategies further optimises the delay and bandwidth consumption during real-time transmission.

3 SCTRT: a sparse coding-based model for vocal audio feature extraction and real-time transmission

The vocal audio feature extraction and compression model, or SCTRT for short, based on sparse coding forms the foundation of this system. By means of effective feature extraction, compression, and transmission, the SCTRT model seeks to overcome the bandwidth problem and the efficient recovery issue of vocal audio signals in real-time transmission. Three key modules make up the model: pre-processing and audio acquisition module; feature extraction and compression module; real-time transmission and recovery module. Though it operates separately, each module collaboratively guarantees short latency from audio acquisition to signal recovery via high efficiency.

3.1 Audio acquisition and pre-processing

Comprising the front-end of the entire SCTRT system, the audio acquisition and pre-processing module's primary responsibility is to pick voice audio signals from the surroundings and do initial processing. Raw audio signal acquisition, denoising, echo cancellation, and pre-emphasis are among the processes in the process. Usually using high-precision microphones or sensors that collect spoken signals in real-time and translate them into digital audio data via analogue-to-digital conversion (ADC), audio acquisition.

Denoising and echo cancellation are especially important since background noise, echoes, and other disturbances usually impair audio signals. While echo cancellation depends on blind source separation (BSS), denoising is typically accomplished with Wiener or adaptive filtering. In the pre-emphasis stage, a high-pass filter increases the high-frequency components in order to reduce the effect of low-frequency components on next feature extraction.

Denoted as x_{pre}, the pre-processed audio stream is:

$$x_{pre} = H(x) = pre\text{-}process(x) \tag{15}$$

The final output x_{pre} is the signal following noise removal, echo cancellation, and pre-emphasis when x is the original collected audio signal and $H(x)$ indicates the pre-processing action.

3.2 Feature extraction and compression

The SCTRT system consists mostly in the module for feature extraction and compression. It compresses sparse features extracted from audio signals already processed using sparse coding to reduce the data size. Sparse coding aims to exhibit the signal as a straight line composed of a few non-zero coefficients by means of dictionary learning. One may depict the sparse coding process by means of the following equation:

$$x_{pre} \approx D\alpha \tag{16}$$

where α is a sparse coefficient reflecting the sparse representation of the signal and D is a dictionary matrix including the basis elements acquired from the audio stream. We minimise the reconstruction error and maintain the coefficients α sparse in the process to derive the ideal sparse representation.

Following the sparse representation, the system compresses it effectively using a suitable method. Certain compression techniques such as Hoffman coding or arithmetic coding reduce the sparse coefficient α into a smaller binary stream. Although the bandwidth is limited, this reduces the volume of data that has to be transmitted and guarantees that the system may effectively transmit the audio data.

The module for real-time transmission picks and delivers the compressed data. This procedure allows the relationship between the redundant information and the compression rate r to be revealed. One can articulate $R_{redundant}$ as:

$$r = f(B_{net}, R_{redundant}) \tag{17}$$

In which case B_{net} represents the network bandwidth. Closely connecting the compression rate to the network bandwidth and redundancy information, the system dynamically

changes these parameters to guarantee optimal performance under various network situations.

3.3 Real-time transmission and recovery

The module in real-time transmission and recovery guarantees that the signal may be precisely recovered at the receiving end and helps to transmit the compressed audio feature data effectively. Using the UDP protocol – which can offer low-latency real-time data transfer – the SCTRT system helps to lower the transmission delay. The UDP protocol does not have an automatic retransmission mechanism, hence packet loss can result quite quickly. FEC allows the SCTRT system to compensate for missing packets. This method sends data with additional information, therefore increasing the dependability of the system.

First decoding the obtained compressed data, the system recovers the sparse coefficients $\hat{\alpha}$ at the receiver side. Subsequently, the sparse coefficients are recovered using the dictionary matrix D as audio signals $\hat{x}$. We then extract the sparse coefficients using the system as audio signals $\hat{x}$.

$$\hat{x} = D\hat{\alpha} \tag{18}$$

By means of this procedure, the receiver can retrieve the original audio signal with low latency and playback it. The system reduces the redundant information to enhance the transmission rate when the network condition is poor and increases the redundant information when the network condition is excellent so further improving the recovery accuracy.

Performance evaluation and system optimisation are absolutely essential to guarantee the great performance of the SCTRT model in several network contexts. The assessment mostly consists in measures of bandwidth efficiency, transmission delay, and audio quality. First computed in the assessment of audio quality, the signal-to-noise ratio (SNR) measures the ratio between the noise and the recovered signal.

$$SNR = 10\log_{10}\frac{\|x\|_2^2}{\|x-\hat{x}\|_2^2} \tag{19}$$

where x represents the original signal, $\|x\|_2^2$ and $\|x-\hat{x}\|_2^2$ respectively indicate the energy of the signal and the recovery error. Greater SNR levels point to improved audio quality.

Furthermore, the following formula computes the error of the recovered signal by means of the mean square error (MSE):

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(x_i - \hat{x}_i\right)^2 \tag{20}$$

where N is the overall number of the sample points of the signals and x_i and $\hat{x}_i$ are the i^{th} sampling points of the original and recovered signals respectively.

In real-time transmission systems, particularly in low-latency application situations, transmission delay is a critical performance indicator. One may represent the transmission delay $T_{latency}$ as:

$$T_{latency} = T_{encode} + T_{transmit} + T_{decode} \tag{21}$$

where T_{encode} is the signal feature extraction and compression process's timing; $T_{transmit}$ is the data transfer time from the transmitter to the receiver; T_{decode} is the time of decoding and recovering the audio signal. Reducing these periods as much as feasible will help to guarantee the system's real-time performance.

We also compute the *compression ratio*, another important assessment statistic as follows:

$$Compression\ ratio = \frac{S_{original}}{S_{compressed}} \tag{22}$$

whereas $S_{compressed}$ and $S_{original}$ respectively are the feature data sizes of the compressed and original audio signals. More information is maintained in the compression process by a higher *compression ratio*, so the technique preserves.

The SCTRT system can be tuned to maximise performance for various evaluation criteria. Typically achieved by dynamically changing the dictionary size and learning rate, optimising for audio quality seeks to improve the accuracy and SNR of signal recovery. Furthermore influencing the recovery accuracy are dictionary update frequency and sparsity constraints; hence, optimising these values will aid to raise the recovered audio quality.

To lower the delay when the network bandwidth is limited while guaranteeing the quality of the recovered audio signal, the system can dynamically change the amount of redundant information depending on the network conditions. System optimisation in terms of transmission delay depends on minimising the time consumed during encoding and decoding as well as on optimising the transmission protocols to lower the data transfer times.

By means of thorough examination and optimisation, the SCTRT system reduces latency, enhances bandwidth efficiency, and guarantees steady performance in several network conditions while so preserving audio quality.

4 Experimental results and analyses

4.1 Datasets

As shown in Tables 1 and 2, this work tested the performance of the model in a range of vocal audio environments and guaranteed the validity of the model by means of two representative vocal audio datasets, VocalSet and PopVocals, with different audio characteristics.

- VocalSet: Covering several genres from classical to contemporary pop, this dataset features several vocal audio techniques. The dataset's audio consists of a range of soloists and choirs spanning thirty seconds to two minutes. Designed with a broad spectrum of pitches, tempos, and dynamic ranges, the dataset is fit for assessing the model's capacity to extract and recover characteristics in many audio settings. Furthermore, the VocalSet's audio features several background noises (e.g., vocal overlay, ambient noise, etc.), which makes it perfect for evaluating SCTRT model audio transmission and recovery in loud surroundings.

- PopVocals: From 20 to 1 minute in length, this dataset comprises vocal clips from popular music spanning a range of well-known pop performers and songs. Covering several pitch shifts, tempo rates, and musical backgrounds, the audio samples in this dataset exhibit a strong sense of rhythm and emotional expression and are fit for testing the resilience of the model in fast changing musical surroundings. Although the PopVocals dataset's audio lacks notable ambient noise, its multi-layered harmonies and background music allow us to investigate how effectively the SCTRT model can interpret many layers of audio in the transmission. At the transmission time, several audio layers exist.

Table 1 VocalSet dataset description

Feature	*Description*
Audio length	30 seconds to 2 minutes
Sampling rate	44.1 kHz
Bit depth	16-bit
Content	Various vocal styles (classical, pop, choral)
Use in experiment	Tests feature extraction with noise and dynamic vocals

Table 2 SSDD dataset scene breakdown

Feature	*Description*
Audio length	20 seconds to 1 minute
Sampling rate	44.1 kHz
Bit depth	16-bit
Content	Pop vocals with rhythmic and emotional expression
Use in experiment	Tests real-time transmission and compression

4.2 Comparative experiments

We have selected in this experiment to evaluate three common audio feature extraction and compression techniques: the MFCC, the STFT, and the conventional sparse coding approach SCTRT with The objective of the experiment is to evaluate, in terms of extracting audio features, file size reduction, restoration of audio quality, and real-time transmission delay computation, the performance of the several approaches MSE, compression ratio, SNR, and latency are four assessment criteria applied in the trials to assess the several approaches' performance.

Two datasets – the VocalSet and PopVocals – containing various kinds of vocal recordings fit for the evaluation of audio compression and transmission were employed in the tests. Every audio sample was cut into 200 ms short time frames to guarantee that the frequency domain characteristics of every segment fairly reflect the whole audio signal. All techniques – including denoising, normalisation, and brief time-frame splitting – use a standard pre-processing step throughout audio signal processing.

First sparse codes the audio signal, then compresses the obtained sparse features to recover the original audio signal using SCTRT model extracting its features. Particularly at larger compression ratios, the SCTRT model is able to keep more features of the audio stream using the sparse coding function. Commonly used in traditional techniques for

audio signal compression and reconstruction is sparse coding. Figures 3 and 4 display the experimental outcomes.

Figure 3 Comparative experimental results on the VocalSet dataset (see online version for colours)

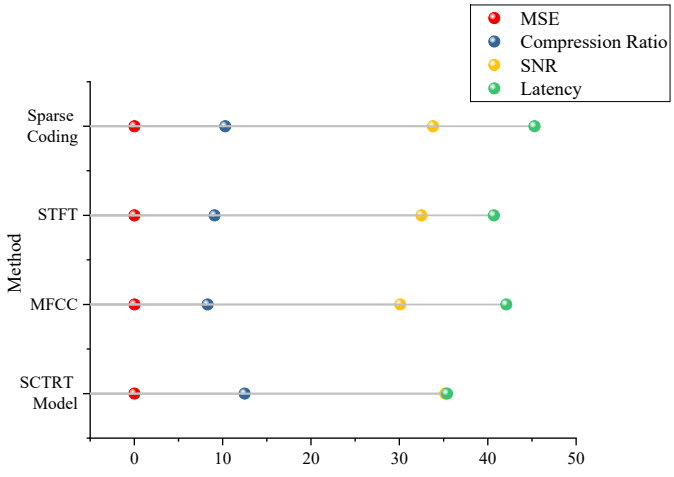
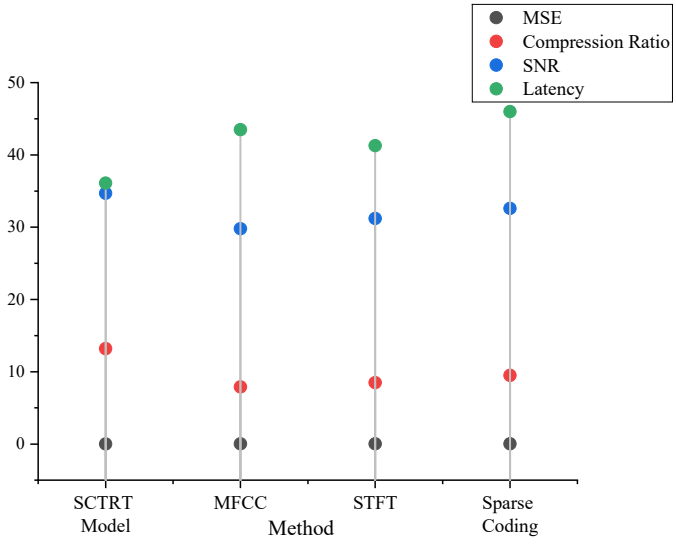


Figure 4 Comparative experimental results on the PopVocals dataset (see online version for colours)



In both datasets the SCTRT model performs better than the other comparison techniques according to the experimental results. First, the SCTRT model clearly shows that it is better in recovering audio signals and can efficiently bring back the original features of audio by having the lowest MSE value. Other techniques, including MFCC and STFT, have greater MSEs, meaning their audio recovery process has more mistakes. Having stated that, notably on the VocalSet dataset, which has the best compression performance,

the SCTRT model boasts a far greater compression ratio than the others. This allows the model to respond to the needs of effective audio transmission by greatly lowering the data amount while preserving the audio quality.

Regarding SNR, the SCTRT model also shows good performance particularly at high compression ratios and keeps a high SNR, so indicating the great capacity of the model to restore audio quality. By comparison, the other techniques sacrifice on the quality of the compressed signal by having reduced SNRs. Finally, the SCTRT model is the ideal option for real-time audio transmission uses requiring low latency and processing in real-time since it boasts the minimum transmission delay.

4.3 Ablation experiments

In this work, we investigated how the several components of the SCTRT model affected the general performance using ablation experiments. We performed the ablation studies by progressively eliminating or substituting several modules and tracking their impacts on measures like compression efficiency, transmission delay, and audio signal recovery quality. Both in terms of pitch, volume, and timbre as well as recordings in a range of acoustic situations, the studies are tested using the VocalSet dataset, which has a wide spectrum of sorts of vocal audio. This variety makes it an excellent test of the generalisation capacity and robustness of the model in a number of complicated audio circumstances in order to fully evaluate the model's performance in actual vocal audio signal processing.

Separately in our tests, we investigated the following situations:

- Original model (SCTRT): Audio acquisition, feature extraction and compression, transmission and recovery modules all are included in the whole model.
- Model with sparse coding module deleted: The usual spectral feature extraction approach is utilised instead once the module of sparse coding feature extraction is deleted.

Removing the feature compression module will preserve the feature extraction and recovery modules according to the model.

Remove the recovery module and test direct transmission following feature compression in the model with removed recovery module.

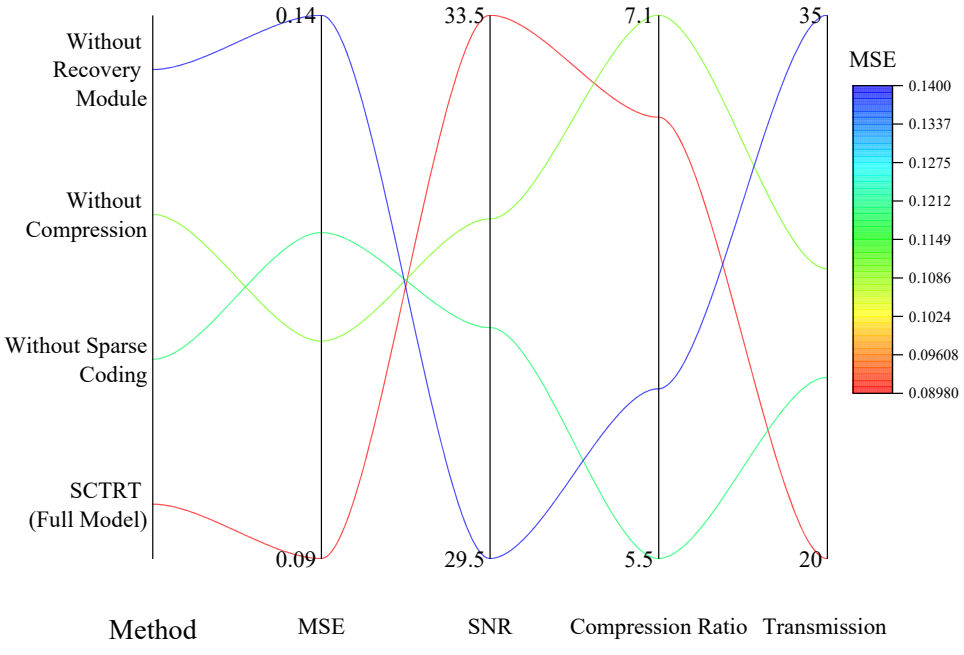
We may thus fairly assess the influence of every module on the general performance of the model by means of comparison of the performance under several configurations. Figure 5 shows the experimental outcomes.

It is clear from eliminating each module how the SCTRT model performed from the ablation studies. First, the original model (SCTRT) exhibits ideal performance based on the MSE of 0.09 dB, SNR (signal to noise ratio) of 33.5 dB, compression ratio of 6.8, and transmission delay remained at roughly 20 ms. This result indicates that the whole approach guarantees high-quality recovery of the audio signal while nevertheless allowing effective compression and low-latency transmission.

Although the MSE and SNR rise to 0.11 dB and 32.0 dB respectively, the compression ratio drops to 7.1, once the feature compression module is removed. This result shows that the elimination of the feature compression module enhances the quality of the signal recovery but compromises the compression efficiency, thus increasing the

amount of communicated data and so influencing the real-time transmission capability of the system.

Figure 5 Results of ablation experiment (see online version for colours)



The ablation studies reveal generally that the SCTRT model consists in integral elements the sparse coding, feature compression, and recovery modules. Particularly with regard to how successfully the model recovers signals and compresses them, the performance of each module declines greatly when one removes them. Eliminating any one module shows varying degrees of performance loss, which highlights even more the dependency and relevance of the modules in the model design.

5 Conclusions

Aiming to accomplish low-latency and high-quality real-time transmission of vocal audio via efficient audio feature extraction and compression approaches, this work proposes a sparse coding-based vocal audio feature extraction and real-time transmission model, SCTRT. By means of comparison and ablation studies, the design and implementation of SCTRT is thoroughly discussed in this paper together with its advantages in terms of audio restoration quality, compression efficiency, and real-time transmission performance.

The SCTRT model has significant restrictions even if it has produced amazing performance in audio feature extraction and real-time transmission. First of all, particularly in the situation of low-quality audio or high background noise, which may not perform well, the model's resilience against noise interference and complicated surroundings has to be strengthened. Second, especially when processing vast amounts of

data, the model's computational complexity is considerable despite effective compression and feature extraction, which could cause rising delay. Furthermore, there is still space for improvement of the real-time transmission performance of the model, particularly in circumstances with limited bandwidth, and it is a significant difficulty to lower the transmission latency and bandwidth usage while preserving audio quality.

One can pursue future studies in the following spheres:

1 Optimising the computational efficiency of the model: Although sparse coding and feature compression take a lot of additional time to perform, especially when processing vast volumes of audio data, the SCTRT model recovers and compresses music better. Future research should investigate more effective sparse coding techniques to lower the computational complexity of the model, such combining deep learning with sparse coding to hasten the feature extraction process or approximative sparse coding.

2 Improving real-time transmission performance: Under the present paradigm, there are certain delay and bandwidth consumption issues during real-time transmission. Future studies can investigate more effective transmission protocols and compression techniques to lower the bandwidth demand and transmission delay of data flow.

3 Multimodal data fusion: Audio signals can be thought of as fused with information from other modalities (e.g., picture, text, etc.), for future joint feature extraction and compression given the broad applicability of multimodal data such as speech, image, video, etc.

Declarations

All authors declare that they have no conflicts of interest.

References

Cui, M. and Prasad, S. (2016) 'Sparse representation-based classification: orthogonal least squares or orthogonal matching pursuit?', *Pattern Recognition Letters*, Vol. 84, pp.120–126.

Cunningham, S. and McGregor, I. (2019) 'Subjective evaluation of music compressed with the ACER codec compared to AAC, MP3, and uncompressed PCM', *International Journal of Digital Multimedia Broadcasting*, Vol. 2019, No. 1, p.8265301.

Deng, M., Meng, T., Cao, J. et al. (2020) 'Heart sound classification based on improved MFCC features and convolutional recurrent neural networks', *Neural Networks*, Vol. 130, pp.22–32.

Gupta, R., Tanwar, S., Tyagi, S. et al. (2019) 'Tactile internet and its applications in 5G era: a comprehensive review', *International Journal of Communication Systems*, Vol. 32, No. 14, p.e3981.

Latif, S., Qadir, J., Qayyum, A. et al. (2020) 'Speech technology for healthcare: opportunities, challenges, and state of the art', *IEEE Reviews in Biomedical Engineering*, Vol. 14, pp.342–356.

Lie, W-N. and Chang, L-C. (2006) 'Robust and high-quality time-domain audio watermarking based on low-frequency amplitude modification', *IEEE Transactions on Multimedia*, Vol. 8, No. 1, pp.46–59.

Michelsanti, D., Tan, Z-H., Zhang, S-X. et al. (2021) 'An overview of deep-learning-based audio-visual speech enhancement and separation', *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 29, pp.1368–1396.

Nasr, M.A., Abd-Elnaby, M., El-Fishawy, A.S. et al. (2018) 'Speaker identification based on normalized pitch frequency and Mel frequency cepstral coefficients', *International Journal of Speech Technology*, Vol. 21, pp.941–951.

Nassra, I. and Capella, J.V. (2023) 'Data compression techniques in IoT-enabled wireless body sensor networks: a systematic literature review and research trends for QoS improvement', *Internet of Things*, Vol. 23, p.100806.

Pawar, M.D. and Kokate, R.D. (2021) 'Convolution neural network based automatic speech emotion recognition using Mel-frequency Cepstrum coefficients', *Multimedia Tools and Applications*, Vol. 80, pp.15563–15587.

Qayyum, A., Saeed Malik, A., Saad, N.M. et al. (2019) 'Image classification based on sparse-coded features using sparse coding technique for aerial imagery: a hybrid dictionary approach', *Neural Computing and Applications*, Vol. 31, pp.3587–3607.

Singh, M.K. (2024) 'A text independent speaker identification system using ANN, RNN, and CNN classification technique', *Multimedia Tools and Applications*, Vol. 83, No. 16, pp.48105–48117.

Umapathy, K., Ghoraani, B. and Krishnan, S. (2010) 'Audio signal processing using time-frequency approaches: coding, classification, fingerprinting, and watermarking', *EURASIP Journal on Advances in Signal Processing*, Vol. 2010, pp.1–28.

Valenzise, G., Prandi, G., Tagliasacchi, M. et al. (2009) 'Identification of sparse audio tampering using distributed source coding and compressive sensing techniques', *EURASIP Journal on Image and Video Processing*, Vol. 2009, pp.1–12.

Wu, D., Hou, Y.T., Zhu, W. et al. (2001) 'Streaming video over the internet: approaches and directions', *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 11, No. 3, pp.282–300.

Xu, C., Sun, C., Jiang, G. et al. (2020) 'Two-level multi-domain feature extraction on sparse representation for motor imagery classification', *Biomedical Signal Processing and Control*, Vol. 62, p.102160.

Yang, M. and De Hoog, F. (2015) 'Orthogonal matching pursuit with thresholding and its application in compressive sensing', *IEEE Transactions on Signal Processing*, Vol. 63, No. 20, pp.5479–5486.