Journal of Computer and Communications, 2025, 13(9), 42-54
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

A Study and Practice of Singing Voice Conversion Based on E-SVS and R-SVC

Frank Ming Han Dong

Hangzhou Foreign Languages School, Hangzhou, China
Email: frank.m.dong@icloud.com

How to cite this paper: Dong, F.M.H. (2025) A Study and Practice of Singing Voice Conversion Based on E-SVS and R-SVC. *Journal of Computer and Communications*, **13**, 42-54.
https://doi.org/10.4236/jcc.2025.139003

Received: August 30, 2025
Accepted: September 19, 2025
Published: September 22, 2025

Abstract

Aiming at the common issues of poor sound quality and significant artifacts involved in today's AI singing voice conversion techniques, this paper proposes a new method of AI-driven singing voice conversion coupled with the development of a deployable system. First, an Expert-based Singing Voice Separation (E-SVS) model based on UVR5 was established to achieve high-fidelity vocal extraction, dereverberation, and denoising by cascading MDX-Net and VR Architecture models. Then, a Retrieval-based Singing Voice Conversion (R-SVC) model is constructed as the core conversion engine. Utilizing HuBERT to extract content features while performing efficient timbre feature retrieval via Faiss, the R-SVC model generated cover audio with highly similar timbre and accurate melody. Finally, by designing a task queue mechanism, WeChat Mini Program front-end and asynchronous processing back-end software is developed to enable providing a smooth user experience, capable of resolving lag issues associated with computationally intensive AI tasks. In practice, it was found that this system can train high-quality models with customized vocals at relatively low data and time costs (10 - 30 minutes of audio).

Keywords

E-SVS, R-SVC, Deep Learning, System Architecture, Asynchronous Processing

1. Introduction

In today's digital content creation field, there is a growing demand for personalized, high-quality music experiences. Nonetheless, traditional music production and cover processes are not only costly but also require professional vocal techniques and recording equipment, excluding the majority of ordinary users and enthusiasts. Meanwhile, most existing AI voice-changing or song generation tools

suffer from issues such as poor output audio quality, mechanical sound, distorted melody, and an inability to produce complex background music. Therefore, such tools are far from creating natural and pleasant "cover" works, having difficulties satisfying the actual expectations of users. The principal issues here are the Singing Voice Separation (SVS) and the Singing Voice Conversion (SVC) technologies.

SVS is the process of deconstructing a complete song mix into its components (such as vocals, drums, bass, etc.). SVC is a technology aiming at transforming the timbre of the source singer's voice to make it sound like the target singer, while crucially retaining the original musical expression elements (such as melody, rhythm, and vocal skills). In contrast with relatively stable speech, singing exhibits vast and rapid dynamic changes in pitch, loudness, and articulation. SVS and SVC are the key technologies for the success of song generation during the process of AI voice-changing song generation. Hence, this paper focuses on the research and development of SVS and SVC technologies.

The current research status of SVS mainly focuses on the fields of multilingual zero-shot and cross-domain style control. For example, the TCSinger2 model proposed by Zhejiang University achieves multi-level style control through natural language text, speech, or singing prompts, and supports zero-shot generation of singing voices in unseen styles [1]. The StyleSinger model achieves Out-of-Domain (OOD) style transfer through the Residual Style Adapter (RSA) and Uncertainty Modeling Layer Normalization (UMLN) technology, focusing mainly on basic style features such as timbre and emotion [2]. In addition, SVS can use speech separation technology and use the model built by Deep Neural Network (DNN) to learn the time-frequency characteristics of different sound source signals, and learn their time-frequency characteristics and spatial structure from the original speech signal data. DNN has a strong ability to model speech signal data, but DNN's ability to model the correlation between different time frames of speech data still cannot meet actual needs [3]. Some scholars have proposed using Recurrent Neural Networks (RNNs) for separation tasks, which can better handle the problem that speech signal data is time-series-related and has long-term and short-term dependencies [4]. Uhlich *et al.* combined data enhancement and other methods to complete the separation task during the training of RNN, and converted it into a two-dimensional image through operations such as short-time Fourier transform to serve as the input of the neural network [5]. Therefore, a large number of studies have also migrated the Convolutional Neural Network (CNN), the most representative algorithm in the field of processing two-dimensional images, to speech separation [6], such as the U-Net [7] architecture in the medical field. Jansson *et al.* used the advantages of U-Net in image segmentation and successfully migrated U-Net to speech separation, realizing the transfer learning of the same network model in different fields [8]. At present, there is no unified standard separation framework at home and abroad, and after separation, signal distortion often occurs, or the predicted target signal contains interference

from the interference signal. The quality of the separated signal needs to be improved.

In recent years, the technology of Singing Voice Conversion (SVC) has made significant progress. A large number of generative models have been able to learn the structural features of music well and output rich music samples. For example, Sequence-to-Sequence (Seq2Seq) models such as Recurrent Neural Networks (RNN) [9] and Transformer [10] are used as music generation methods to encode music sequences and emotion labels, learn the feature representation of specific emotions, and then decode the feature representation to generate music with corresponding emotions, or control the generated music with the assistance of music emotion classifiers. Recent research, Music FaderNets [11] uses Gaussian Mixture Variational Autoencoders (GMVAE) [12] to learn the latent space representation of note features and rhythm features on the Arousal dimension of Russell's two-dimensional emotion narrow space [13] through a semi-supervised method, and uses these two features to achieve control of the Arousal dimension, thereby generating music with specific emotions. Most existing music generation models use Recurrent Neural Networks (RNN) as encoders and decoders [14] [15]. Music is a long sequence of data, and RNN has limited modeling capabilities for such data. It is easy to lose contextual dependencies and suffer from the problem of gradient vanishing or gradient exploding.

The main contribution of this paper is the design and development of an innovative and deployable AI-driven singing voice conversion and generation system. This system accepts any songs uploaded by users and precisely substitutes the specified target vocals for the original vocals, generating a "cover" song performed by the new voice that is both musically harmonious and natural-sounding.

1) A novel multi-stage Expert-based Singing Voice Separation (E-SVS) system was devised and validated. By decomposing the complex SVS task into three subtasks of extraction, dereverberation, and denoising, and assigning optimized dedicated models to each, the result demonstrates that the system possesses superior separation performance and artifact control compared to single and end-to-end general separation models.

2) Using Retrieval-based Singing Voice Conversion (R-SVC), the system can generate cover audio with highly similar timbre, accurate melody, and minimal artifacts, sounding far better than traditional general models.

3) Efficient model customization: Taking advantage of pre-training, R-SVC models can complete the training of a high-quality model with a small amount, 10 - 30 minutes, of audio data.

4) We implemented a robust and extensible asynchronous processing backend architecture designed specifically for computationally intensive AI audio tasks. This architecture fully decouples front-end requests from long-term back-end computations via a task queue mechanism, ensuring instantaneous user interface responsiveness and a smooth user experience. This provides a reusable architectural paradigm for the productization of similar AI applications.

2. Establishment of a Multi-Stage Expert-Based Singing Voice Separation (E-SVS) Model

Singing Voice Separation (SVS) is the process of deconstructing a complete song mix into its components (such as vocals, drums, bass, etc.), aiming at losslessly separating lead vocals from accompaniment. As a preprocessing step, the separation quality of SVS directly determines the purity of the subsequent SVC model input, which in turn has a decisive impact on the "cover" quality of the final output. Any residual accompaniment artifacts or vocal impairments will be brought in and potentially amplified during the conversion process. As a result, a multi-stage Expert-based Singing Voice Separation (E-SVS) model is proposed and established.

The multi-stage E-SVS system utilizes a three-step separation method implemented with the open-source Ultimate Vocal Remover 5 (UVR5). UVR5 integrates multiple industry-leading separation algorithms, allowing for a modular processing pipeline. This strategy decomposes the complex SVS task into three subtasks, assigning each an optimized dedicated model to achieve separation results that far exceed those of a general model. The detailed structure of this three-step processing pipeline is shown in Figure 1. By using three specialized models connected in series (UVR-MDX-NET-Voc_FT, UVR-De-Echo-Normal, UVR-DeNoise), it gradually extracts studio-quality dry vocals from the raw audio. The specific steps are as follows.

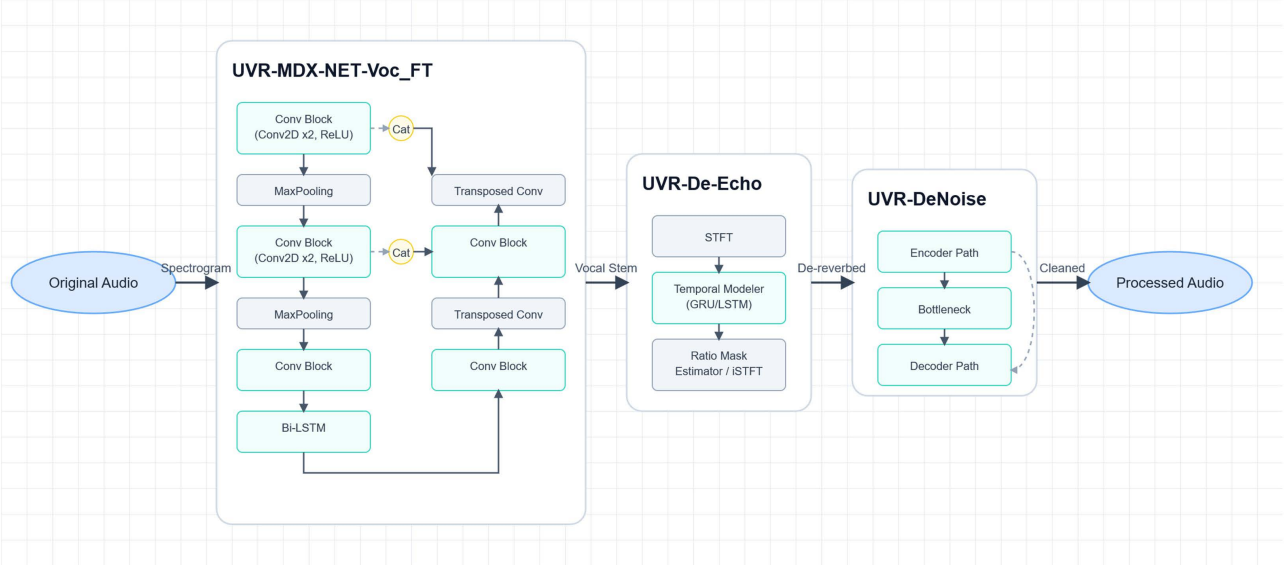

Figure 1. SVS process for Expert-based Singing Voice Separation (E-SVS).

Step 1: Main Vocal Extraction (MDX-Net)

As shown on the left side of Figure 1, the first step in the pipeline is to select the MDX-Net architecture and fine-tune the UVR-MDX-NET-Voc_FT model specifically for vocal extraction. By using a branch operating in the time domain (e.g., a convolutional network based on U-Net) and a branch operating in the fre-

quency domain (e.g., a spectrogram obtained from a short-time Fourier transform), MDX-Net can simultaneously capture both the macroscopic waveform structure (e.g., transients in percussive instruments) and the microscopic spectral features (e.g., the harmonic structure of vocals) of audio signals. This dual-stream design is highly effective for accurately isolating specific sound sources (e.g., vocals) from complex music mixes. This allows for a preliminary separation of a relatively complete and dry vocal track from the original mix.

Step 2: Echo and Reverberation Removal (VR Architecture)

As shown in the middle of Figure 1, this step focuses on removing echo and reverberation. The model used is the UVR-De-Echo-Normal module of the VR Architecture. Its primary goal is to address the residual spatial acoustic effects in the vocal output from the first step, thereby eliminating the echo and reverberation commonly found in commercial music. This results in a studio-quality "dry vocal" and prevents acoustic artifacts from being fed into the E-SVC model.

Step 3: Fine-grained Denoising (VR Architecture)

As shown on the right side of Figure 1, the UVR-DeNoise module from the VR Architecture is selected as the model. As a final refinement step, this model identifies and removes non-musical background noise from the vocal track, such as hiss, electrical hum, and other ambient noise. This step ensures that the final vocal track fed into the E-SVC model is as pure as possible, providing optimal input conditions for high-quality voice conversion.

To visually demonstrate the effectiveness of this three-step method, we conducted a spectral analysis of its processing on a dense, complex real-world audio sample. The results are shown in Figure 2. The extremely dense spectrum of the original mix (Figure 2(a)) reveals that the vocal signal is obscured by numerous instruments and strong transients (represented by bright vertical lines running through the spectrum). After the first step of MDX-Net processing (Figure 2(b)), the spectral complexity is significantly reduced, reflecting that the accompaniment has been largely separated, allowing the harmonic structure of the vocal to become clearly visible for the first time. Finally, after subsequent dereverberation and denoising, we obtain the clean, dry audio signal shown in Figure 2(c): an audio signal with a clean background and only the clear harmonics of the vocal. This visual transformation from a "sound wall" to a pure vocal sharply demonstrates the ability of the E-SVS pipeline to gradually refine the signal and validates its rationale and necessity as a high-quality input for the subsequent R-SVC module.

3. Establishment of the Retrieval-Based Singing Voice Conversion Theory Model R-SVC

The essence of Singing Voice Conversion (SVC) lies in modifying the vocal characteristics of a source singer to those of a target singer while precisely preserving the original language content, melodic contours, rhythm, and vocal techniques. This high requirement for musicality makes SVC more challenging than general

speech conversion. Therefore, a Retrieval-based Singing Voice Conversion (R-SVC) was developed. By adopting a hybrid "Retrieval-Generation" paradigm, it strikes a balance between conversion quality, training efficiency, and ease of use.

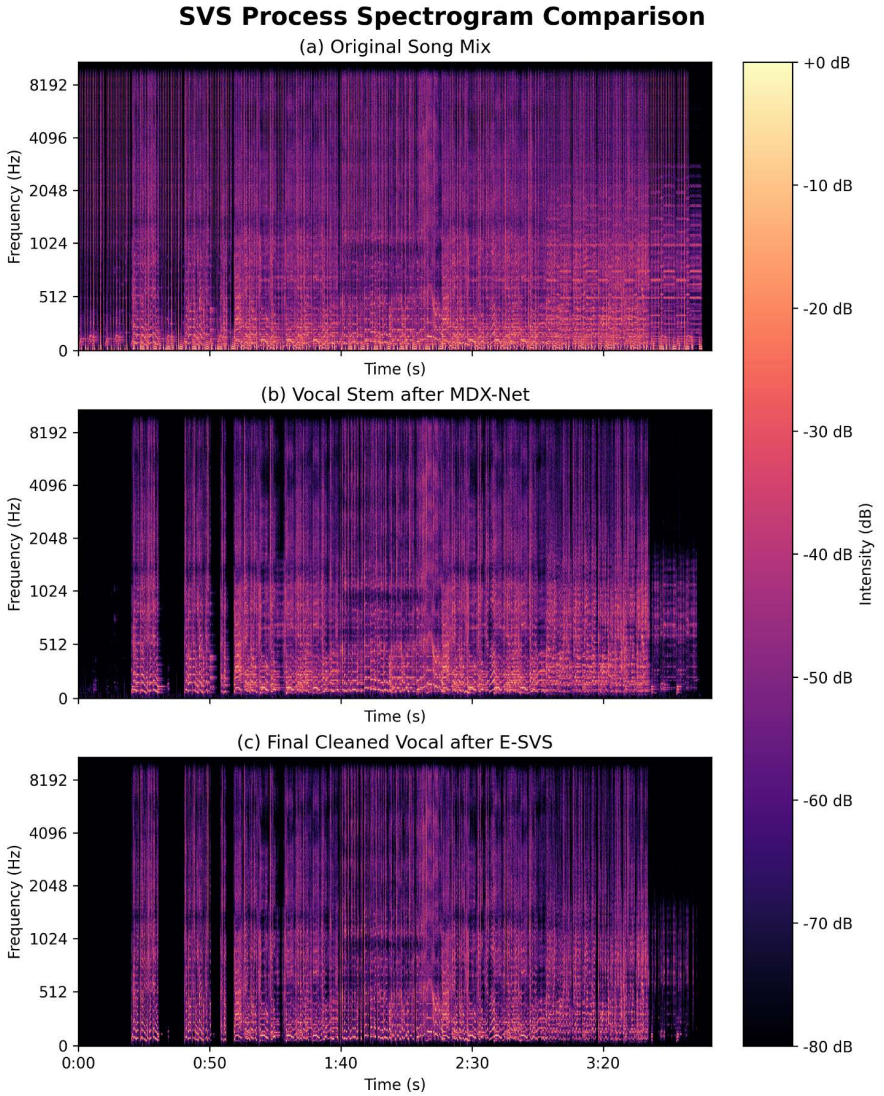

Figure 2. Spectral comparison of the separation performance of the E-SVS process on real, high-density audio samples. (a) shows the spectrum of the original song mix, demonstrating its dense signal density and complex structure. (b) shows the vocals after preliminary extraction using MDX-Net, with most of the accompaniment removed. (c) shows the final, dry vocals after complete dereverberation and denoising, showing a clean background and clear harmonic structure. This comparison strongly demonstrates the effectiveness of the multi-stage E-SVS process employed in this study.

SVC Core Architecture and Principles: R-SVC integrates multiple pre-trained components. Its workflow is shown in **Figure 3**. The left side of the process is a one-time, offline model training phase for learning the target timbre; the right side is a real-time, online voice conversion (inference) phase for generating cover songs.

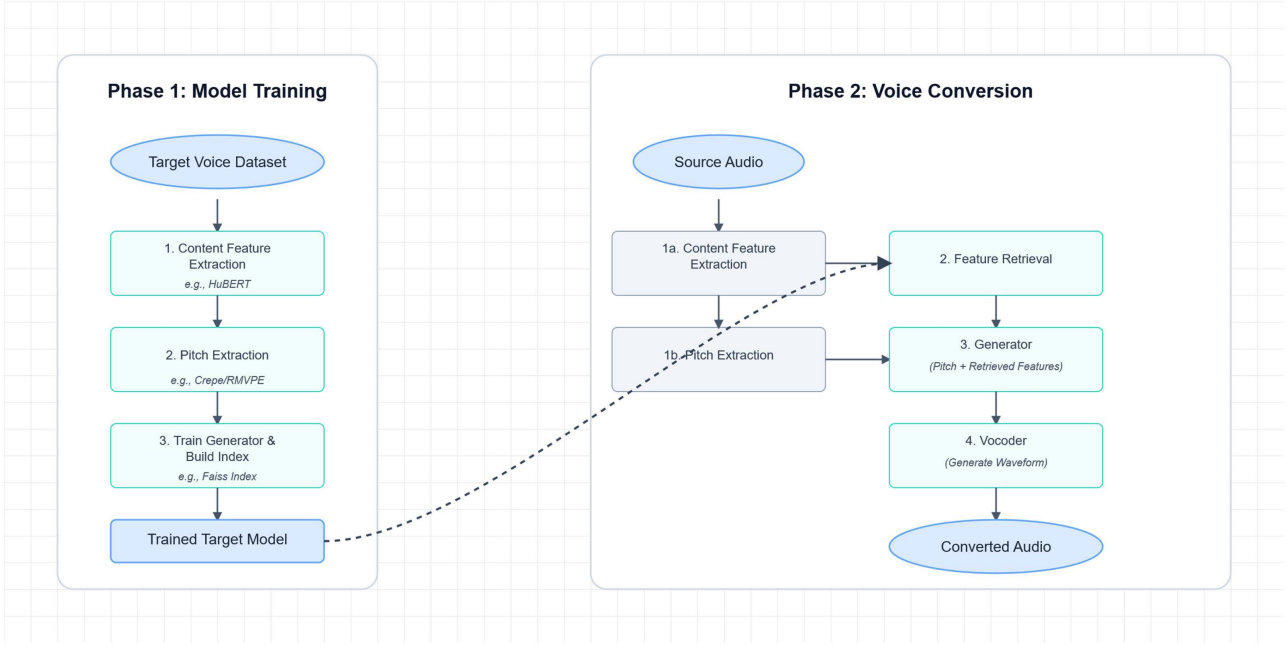

Figure 3. RVC workflow and core architecture diagram.

As shown in the "Model Training" stage on the left side of **Figure 3**, R-SVC's efficiency stems from its unique pre-training and indexing mechanisms. First, developers provide a dataset of high-quality dry audio of a target singer. The process extracts content features from this audio using a pre-trained model (HuBERT) and obtains pitch information using a specialized pitch extraction algorithm (such as Crepe/RMVPE). The system then uses these features to train a generator model and simultaneously constructs a feature index for an efficient vector search engine (such as Faiss). This process results in a lightweight "trained target model" containing specific timbre features, ready for subsequent voice conversion.

The core steps of voice conversion (as shown on the right side of **Figure 3**) are as follows:

Step 1. Content & Pitch Feature Extraction: First, the system extracts two core pieces of information in parallel from the clean source vocals processed by E-SVS. Using a powerful pre-trained Self-Supervised Learning (SSL) model, HuBERT (Hidden-Unit BERT), it extracts content-related acoustic features that are independent of timbre. Pre-trained on a large, unlabeled speech dataset, HuBERT learns to encode speech signals into a highly condensed representation that is insensitive to speaker identity. These features capture both "what" (linguistic content) and "how" (melody, rhythm) of the vocals, but strip away the original singer's timbre. The HuBERT model was selected as the content encoder due to its state-of-the-art performance in extracting speaker-invariant linguistic features from raw audio, which is crucial for achieving an effective decoupling of content from timbre [16]. Simultaneously, the system extracts the source vocal's pitch contour, which will be used to guide the subsequent generation process. For this task, the RMVPE algorithm was chosen, as it provides high accuracy for complex

singing pitch contours while efficiently leveraging GPU acceleration [17].

Step 2. Feature Retrieval: Next, RVC utilizes a feature index library specifically built for the target singer's voice. This library stores timbre features extracted from a small amount of high-quality dry vocal data of the target singer (usually only 10-30 minutes). Once the content features of the source vocals are extracted, R-SVC uses an efficient vector retrieval engine called Faiss to perform a fast similarity-driven search within the target timbre library. It finds the timbre segments in the target library that best match the current source content features. This "retrieval" step is the core innovation of R-SVC. Rather than generating timbre out of thin air, it "borrows" the most relevant parts from real target timbre samples, which greatly enhances the realism and detailed texture of the generated sound.

Step 3. Feature Fusion & Generation: In this stage, the model intelligently fuses the pitch features of the source voice (from step 1) with the timbre features retrieved from the target library (from step 2). This fused feature representation is fed into a generator neural network to produce an abstract acoustic representation (such as a mel-spectrogram).

Step 4. Waveform Synthesis: Finally, a vocoder is responsible for converting the abstract acoustic feature representation generated in the previous step back into an actual, audible audio waveform. R-SVC typically employs GAN-based vocoders (such as the BigVGAN variant integrated into the project itself), which are known for generating high-fidelity, artifact-free audio [18]. The final output waveform retains the melody and content of the source song while imbuing it with the unique timbre of the target singer.

4. System Development and Implementation

By integrating the UVR5-based three-step E-SVS model and the Retrieval-based Singing Voice Conversion (R-SVC), an AI-supported singing voice generation system was developed based on R-SVC model training. The back-end service was implemented in Python, complementing the WeChat Mini Program front-end.

4.1. Training and Inference of R-SVC Model

The R-SVC model was trained using the open-source Retrieval-based-Voice-Conversion-WebUI (RVC) v2 on a single NVIDIA GeForce RTX 4090 GPU. The E-SVS pipeline described in this study was implemented using Ultimate Vocal Remover 5 (UVR5).

The datasets used in this study consist of a training set and a test set. The target voice dataset used for training the R-SVC model contains approximately 56 minutes of dry vocal R&B tracks from a single male singer. Specifically, all audio files are monophonic WAV files with a sampling rate of 44.1 kHz. The test set comprises full-song dry vocals, processed by the E-SVS model described previously, from another male singer, to assess the model's generalization capabilities. These tracks are also in the R&B genre and were standardized as monophonic 44.1 kHz WAV files.

The key hyperparameters used for the training process are detailed in Table 1. The entire training process for the 56-minute dataset took approximately 20 minutes to complete.

Table 1. Key training hyperparameters for the R-SVC model.

Hyperparameter	Value
Target Sample Rate	40k
Pitch Guidance	True
Total Training Epochs	20
Batch Size per GPU	4
Save Frequency	Every 5 epochs
Pitch Extraction Algorithm	rmvpe_gpu
CPU Processes	14

To generate the final converted audio for evaluation and for the end-user application, a specific set of inference parameters was utilized. These parameters, detailed in Table 2, were optimized through experimentation to achieve a balance between timbre similarity, audio quality, and artifact suppression.

Table 2. Key inference (voice conversion) parameters for the R-SVC model.

Parameter	Value	Description
Pitch Extraction Algorithm	rmvpe	High-quality pitch results with low GPU requirements.
Search Feature Ratio	0.75	Controls the accent strength of the conversion.
Volume Envelope Scaling	0.25	Partially mimics the source volume to create natural dynamics.
Voiceless Consonant Protection	0.33	Protects voiceless consonants and breath sounds to prevent artifacts.
F0 Median Filtering Radius	3	Applies median filtering to the pitch results to reduce breathiness.

4.2. Development of Front-End WeChat Mini Program

WeChat Mini Programs were chosen as the front-end platform based on their massive user base in the Chinese market and their ready-to-go convenience, requiring no installation. Development leverages WeChat's native APIs, with the following functional modules:

1) File Upload: Using the wx.chooseMessageFile API, users can conveniently select and upload audio files from WeChat chat logs or their phone's local storage.

2) Network Request: Using wx.request to communicate with the backend API, file uploads and status polling are implemented.

3) User Feedback: Using wx.showLoading to display a full-screen, non-cancellable loading indicator at the start of processing, and using wx.showToast to pro-

vide brief feedback upon completion or failure.

4) Audio Playback: Using the InnerAudioContext API to create an audio instance, the converted song can be played, paused, stopped, and progressed.

4.3. Asynchronous Processing Architecture Design for Backend Service Software

AI model inference, especially audio processing, is a computationally intensive task, typically taking seconds to minutes. Using a traditional synchronous request-response model would block the WeChat Mini Program frontend for extended periods, resulting in a lag and a poor user experience. Therefore, we designed an asynchronous processing architecture, incorporating the Celery task queue system. The entire backend workflow follows this asynchronous task processing model, ensuring immediate frontend responses and stable backend task execution:

Step 1. API Request Received: The WeChat Mini Program frontend initiates a POST request to the backend API, containing the user-uploaded audio file and the ID of the selected target timbre model.

Step 2. Task Enqueue and Immediate Response: Upon receiving the request, the FastAPI application does not immediately perform the time-consuming AI processing. Instead, it packages the processing task (including audio data and parameters) and pushes it as a job to the Celery task queue. The API then immediately returns a 202 Accepted status code and a unique task_id to the Mini Program. This process is extremely fast, with virtually no latency noticeable on the Mini Program side.

Step 3. Background Worker Execution: One or more independent, long-running Python worker processes (workers) continuously monitor the task queue. Upon discovering a new task, an idle worker removes the task from the queue and begins executing the complete E-SVS to R-SVC processing pipeline in the background. This process is completely decoupled from the main API service, so even if the processing time is long, it will not affect the API's ability to receive new user requests.

Step 4. Status Query and Result Retrieval: After receiving the task_id, the mini-program frontend starts a timer and initiates a GET request to another status query API on the backend every few seconds. The backend queries the task queue for the corresponding job based on the task_id and returns the status to the frontend.

Step 5. Task Completion: When the worker completes the audio conversion, it stores the resulting file on the server and associates the result URL with the "Completed" status associated with the task_id. The next time the frontend polls for status, it returns the "Completed" status and the URL of the resulting file, allowing the frontend to download and play the converted song.

In order to achieve collaborative work between the front-end and back-end, the key endpoints of the back-end API are designed and determined, as detailed in Table 3.

Table 3. Key API Endpoints for Front-End and Back-End Coordination.

Function	HTTP Method	Endpoint	Payload	Success Response	Notes
Initiate a conversion task	POST	/api/v1/convert	multipart/form-data: audio_file (file), model_id (string)	202 Accepted, JSON: {"task_id": "unique_id_string"}	Return the task ID immediately, without waiting for processing to complete.
Query task status	GET	/api/v1/status/{task_id}	N/A	200 OK, JSON: {"status": "processing/completed/failed", "result_url": "url_or_null"}	The frontend polls this interface to get the latest status.
Get model list	GET	/api/v1/models	N/A	200 OK, JSON: [{"id": "model1", "name": "Singer A "}, …]	Used to display available target timbres on the front end dynamically.

5. Conclusions and Outlook

This article describes the design, development, and implementation of an AI-driven singing voice conversion product. We developed an end-to-end AI singing voice conversion prototype system, addressing the full engineering challenges from original song input to high-quality cover output.

1) By integrating a refined singing voice separation process, the UVR5-based three-step E-SVS model, and the efficient singing voice conversion model R-SVC, the system achieves high-quality timbre replacement while preserving musicality and is packaged in a user-friendly WeChat mini-program.

2) Not only was a processing pipeline designed to maximize the preservation of musical information, but a robust asynchronous backend architecture was also designed during development to address user experience issues caused by high latency in the AI model.

While the system is effective, it has several technical limitations that point to directions for future work. The conversion quality can degrade in challenging scenarios, such as cross-gender conversions with large pitch range differences (e.g., male bass to female soprano), where it may introduce audible artifacts. The system is also highly sensitive to the quality of the input audio; noisy user uploads can lead to a significant drop in output quality as the E-SVS module may fail to produce a perfectly clean vocal stem. Future technical improvements will therefore focus on improving the robustness of the SVS pipeline and exploring few-shot or zero-shot adaptation techniques to enhance cross-gender conversion performance.

Beyond the technical scope, many issues warrant further analysis and research, such as the right of sound and publicity and the ownership of training data. Using copyrighted songs or sound recordings to train AI models may constitute copyright infringement. To mitigate these risks, the R-SVC model training used in this article will strictly require users to use sound data they own the rights to or have explicitly authorized, as outlined in its terms of service.

Acknowledgements

The development of this research benefited from several excellent open-source projects. Special thanks to the Ultimate Vocal Remover 5 (UVR5) and Retrieval-based-Voice-Conversion-WebUI (RVC) projects and their development communities for providing excellent tools and code.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Zhang, Y., Guo, W., Pan, C., Yao, D., Zhu, Z., Jiang, Z., *et al.* (2025) Tcsinger 2: Customizable Multilingual Zero-Shot Singing Voice Synthesis. *Findings of the Association for Computational Linguistics*: *ACL* 2025, Vienna, 13280-13294. https://doi.org/10.18653/v1/2025.findings-acl.687

[2] Zhang, Y., Huang, R., Li, R., He, J., Xia, Y., Chen, F., *et al.* (2024) Stylesinger: Style Transfer for Out-of-Domain Singing Voice Synthesis. *Proceedings of the AAAI Conference on Artificial Intelligence*, **38**, 19597-19605. https://doi.org/10.1609/aaai.v38i17.29932

[3] Fan, Z., Jang, J.R. and Lu, C. (2016) Singing Voice Separation and Pitch Extraction from Monaural Polyphonic Audio Music via DNN and Adaptive Pitch Tracking. 2016 *IEEE Second International Conference on Multimedia Big Data* (*BigMM*), 20-22 April 2016, 178-185. https://doi.org/10.1109/bigmm.2016.56

[4] Li, P., Yang, Y.W., Gao, X.J., *et al.* (2020) A Study of Chinese Speech Recognition Based on Bidirectional Recurrent Neural Network. *Journal of Applied Acoustics*, **39**, 464-471.

[5] Uhlich, S., Porcu, M., Giron, F., Enenkl, M., Kemp, T., Takahashi, N., *et al.* (2017) Improving Music Source Separation Based on Deep Neural Networks through Data Augmentation and Network Blending. 2017 *IEEE International Conference on Acoustics*, *Speech and Signal Processing* (*ICASSP*), New Orleans, 5-9 March 2017, 261-265. https://doi.org/10.1109/icassp.2017.7952158

[6] Tan, K., Chen, J. and Wang, D. (2018) Gated Residual Networks with Dilated Convolutions for Supervised Speech Separation. 2018 *IEEE International Conference on Acoustics*, *Speech and Signal Processing* (*ICASSP*), Calgary, 15-20 April 2018, 21-25. https://doi.org/10.1109/icassp.2018.8461819

[7] Ronneberger, O., Fischer, P. and Brox, T. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells, W. and Frangi, A. Eds., *Lecture Notes in Computer Science*, Springer International Publishing, 234-241. https://doi.org/10.1007/978-3-319-24574-4_28

[8] Jansson, A., Humphrey, E., Montecchio, N., *et al.* (2017) Singing Voice Separation with Deep U-Net Convolutional Networks. *Proceedings of the* 18*th International Society for Music Information Retrieval Conference*, Montreal, 23-27 October 2017, 23-27.

[9] Tie, Y., Chen, H.J., Jin, C., *et al.* (2022) Research on Emotion Recognition Method Based on Audio and Video Feature Fusion. *Journal of Chongqing University of Technology* (*Natural Science*), **36**, 120-127.

[10] Sulun, S., Davies, M.E.P. and Viana, P. (2022) Symbolic Music Generation Conditioned on Continuous-Valued Emotions. *IEEE Access*, **10**, 44617-44626.

https://doi.org/10.1109/access.2022.3169744

[11] Tan, H.H. and Herremans, D. (2020) Music FaderNets: Controllable Music Generation Based on High-Level Features via Low-Level Feature Modelling. *Proceedings of 21st International Society for Music Information Retrieval Conference* (*ISMIR*), Montréal, 11-16 October 2020, 109-116.

[12] Dilokthanakul, N., Mediano, P.A.M., Garnelo, M., *et al.* (2017) Deep Unsupervised Clustering with Gaussian Mixture Variational Autoencoders. *International Conference on Learning Representations* (*ICLR*), Toulon, 24-26 April 2017, 1-12.

[13] Russell, J.A. (1980) A Circumplex Model of Affect. *Journal of Personality and Social Psychology*, **39**, 1161-1178. https://doi.org/10.1037/h0077714

[14] Roberts, A., Engel, J., Raffel, C., *et al.* (2018) Ahierarchical Latent Vector Model for Learning Long-Term Structure in Music. *International Conference on Machine Learning* (*ICML*), Stockholm, 10-15 July 2018, 4364-4373.

[15] Xu, B. and Liu, T. (2024) Semi-Supervised Emotional Music Generation Method Based on Improved Gaussian Mixture Variational Autoencoders. *Computer Science*, **51**, 281-296.

[16] Hsu, W., Bolte, B., Tsai, Y.H., Lakhotia, K., Salakhutdinov, R. and Mohamed, A. (2021) Hubert: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, **29**, 3451-3460. https://doi.org/10.1109/taslp.2021.3122291

[17] Wei, H., Cao, X., Dan, T. and Chen, Y. (2023) RMVPE: A Robust Model for Vocal Pitch Estimation in Polyphonic Music. ArXiv Prepint.

[18] Sanggil, L., Wei, P., Boris, G., *et al.* (2022) BigVGAN: A Universal Neural Vocoder with Large-Scale Training. ArXiv Prepint.