TimbreSense: Timbre Abnormality Detection for Bel Canto with Smart Devices

YUZHENG ZHU, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
CHENGZHE LUO, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
YONGPAN ZOU, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
DONGPING CHEN, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
KAISHUN WU, Information Hub, The Hong Kong University of Science and Technology - Guangzhou Campus, Guangzhou, China

With the rise of mobile devices, bel canto practitioners increasingly utilize smart devices as auxiliary tools for improving their singing skills. However, they frequently encounter timbre abnormalities during practice, which, if left unaddressed, can potentially harm their vocal organs. Existing singing assessment systems primarily focus on pitch and melody and lack real-time detection of bel canto timbre abnormalities. Moreover, the diverse vocal habits and timbre compositions among individuals present significant challenges in cross-user recognition of such abnormalities. To address these limitations, we propose TimbreSense, a novel bel canto timbre abnormality detection system. TimbreSense enables real-time detection of the five major timbre abnormalities commonly observed in bel canto singing. We introduce an effective feature extraction pipeline that captures the acoustic characteristics of bel canto singing. By applying temporal average pooling to the Short-Time Fourier Transform spectrogram, we reduce redundancy while preserving essential frequency-domain information. Our system leverages a transformer model with self-attention mechanisms to extract correlation and semantic features of overtones in the frequency domain. Additionally,

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants 62172286 and U2001207, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2022A1515011509, in part by the Guangdong Provincial Key Laboratory of Integrated Communication, Sensing, and Computation for Ubiquitous Internet of Things under Grant 2023B1212010007, in part by the Project of the DEGP under Grant 2023KCXTD042, and by Tencent's "Rhinoceros Birds" Scientific Research Fund for Young Researchers of Shenzhen University.
Authors' Contact Information: Yuzheng Zhu, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong, China; e-mail: zhuyuzheng2022@email.szu.edu.cn; Chengzhe Luo, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong, China; e-mail: luochengzhe2020@email.szu.edu.cn; Yongpan Zou (Corresponding author), College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong, China; e-mail: yongpan@szu.edu.cn; Dongping Chen, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong, China; e-mail: 3265637790@163.com; Kaishun Wu, Information Hub, The Hong Kong University of Science and Technology - Guangzhou Campus, Guangzhou, Guangdong, China; e-mail: wuks@hkust-gz.edu.cn.

ACM 1550-4859/2025/01-ART7
https://doi.org/10.1145/3708545

we employ a few-shot learning approach involving pre-training, meta-learning, and fine-tuning to enhance the system's cross-domain recognition performance while minimizing user usage costs. Experimental results demonstrate the system's strong cross-user domain recognition performance and real-time capabilities.

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing**; *Ubiquitous and mobile computing systems and tools*; Ubiquitous and mobile devices;

Additional Key Words and Phrases: Bel canto, timbre abnormality, few-shot learning, meta learning

ACM Reference Format:
Yuzheng Zhu, Chengzhe Luo, Yongpan Zou, Dongping Chen, and Kaishun Wu. 2025. TimbreSense: Timbre Abnormality Detection for Bel Canto with Smart Devices. *ACM Trans. Sensor Netw.* 21, 1, Article 7 (January 2025), 20 pages. https://doi.org/10.1145/3708545

1 Introduction

Timbre is a key indicator of a singer's vocal quality, shaped by vocal production organs such as the vocal cords, throat, and nasal passages. Improper vocal techniques can cause various timbre abnormalities, leading to less pleasing performances and potentially harming the singer's vocal health [22, 25]. For instance, excessive tension in the vocal cords during phonation and improper muscle use can cause phonotraumatic vocal hyperfunction. Once formed, these lesions may hinder proper vocal fold closure, reducing sound production efficiency [23, 26]. In traditional bel canto training, experienced instructors quickly identify students' pronunciation issues, enabling them to sense and correct improper vocal exertion. This approach deepens students' understanding while preventing harm caused by incorrect pronunciation. However, this traditional method relies heavily on the expertise of the instructors and remains subjective. Moreover, hiring professional vocal instructors can create an unaffordable financial burden for learners.

With the development of sensing technology, different modalities have been widely used in various sensing tasks, such as health monitoring [41, 42] and identity authentication [45, 46]. An increasing number of intelligent systems are also available to assist in vocal practice. Most commercial intelligent singing training systems focus on melody and rhythm [10, 44], overlooking the crucial element of timbre perception and thus failing to meet the needs of bel canto practice. Other singing voice evaluation techniques are limited by specific experimental conditions [27, 29, 39], such as particular timbre categories, specific singers, controlled environments, and professional equipment. Due to these limitations, no methods currently exist for real-time detection of bel canto timbre abnormalities using portable devices like smartphones.

The types of timbre abnormalities mainly include the following five categories [6, 24, 38]:

- Too bright sound (*White*): excessive brightness may result from limited pharyngeal space or tension in the pharyngeal resonator. This sound is perceived as rough, dry, straight, and lacking resonance.
- Tongue inward (*Uvular*): produced by inward hyoid contraction and tongue root retraction, this sound lacks color and penetration. It is perceived as stiff and overly dark.
- Hypo-nasality (*Nasal*): hypo-nasality makes the singer sound "stuffy" or as if they have a blocked nose. A low uvula and soft palate cause nasal obstruction during sound production, leading to improper nasal resonance.
- Breathy sound (*Breathy*): poor vocal cord closure results in a sound accompanied by noticeable breathy noise, perceived as weak, dull, and hoarse.
- Raised larynx (*Larynx-lift*): excessive throat constriction and a raised larynx produce a squeezed, harsh sound, sometimes leading to a straight tone.

In acoustics, singing timbre is primarily determined by the number, selection, and strength of overtones [37]. This factor significantly contributes to individual differences in timbre. During singing, the vocal cords vibrate to produce the fundamental tone. The fundamental tone resonates within the vocal cavities, generating overtones. These overtones continue to resonate, producing higher-frequency overtones. Consequently, overtones exhibit semantic features across various frequency bands and correlations among themselves.

In summary, there are three challenges in our work. First, variations in vocal structure and singing techniques cause sounds within the same timbre abnormality category to differ acoustically. This challenge impacts the model's generalization, as shown in Section 5.4. Second, the interrelation among overtones in the frequency domain makes it essential to capture their correlation and semantic features in the spectrogram. Third, inexperienced trainees often struggle with muscle control during bel canto practice, leading to unpredictable timbre abnormalities. Post-practice feedback makes it hard for trainees to recall when errors occurred. In contrast, real-time monitoring helps them remember the muscle states linked to incorrect phonation, aiding in error avoidance and enhancing learning. Additionally, bel canto practice sessions last 1 to 2 hours, and failing to correct errors promptly can harm the throat. Therefore, the system needs to be real-time and deployable on mobile devices. In this article, we develope a real-time timbre abnormality detection system for mobile devices, offering immediate feedback to bel canto trainees. We propose a deep learning model using average-pooled spectral features and a self-attention mechanism, tailored specifically to bel canto singing. The model reduces parameters and computational complexity while maintaining effective recognition. We also introduce a meta-learning approach for cross-user training. After pre-training, we customize task generation strategies for our specific task in meta-learning. Finally, the model requires minimal target user data for fine-tuning to achieve strong generalization.

The contributions of our work are as follows:

- To the best of our knowledge, we are the first to propose a timbre abnormality detection system tailored for bel canto, evaluating five types of abnormalities and deploying it on a mobile platform.
- We adaptively modify the input of the transformer model to better suit our task, efficiently extracting correlation and semantic features of overtones in the frequency domain with fewer parameters, ensuring real-time performance.
- We design few-shot learning task generation strategies tailored to our task, integrating pre-training, meta-learning, and fine-tuning. Compared with other classical cross-domain methods, our approach achieves the best results in this application.

The rest of this article is as follows. Section 2 introduces related work on singing evaluation. Section 3 presents the system design. Section 4 provides details on dataset collection and model deployment. Section 5 covers the evaluation. Section 6 is the discussion. Section 7 concludes the article.

2 Related Work

2.1 Acoustic Features of Timbre

Several studies have investigated the link between timbre and acoustic features. Acoustic feature analysis uses signal processing techniques to extract features from audio. These features include the singing power ratio [3, 16], fundamental frequency perturbations [29], average energy [25], and resonance peak correlations [19]. Experts then analyze these features statistically to establish correlations between specific timbre concepts and acoustic characteristics [1, 11, 18, 19, 25, 29]. For

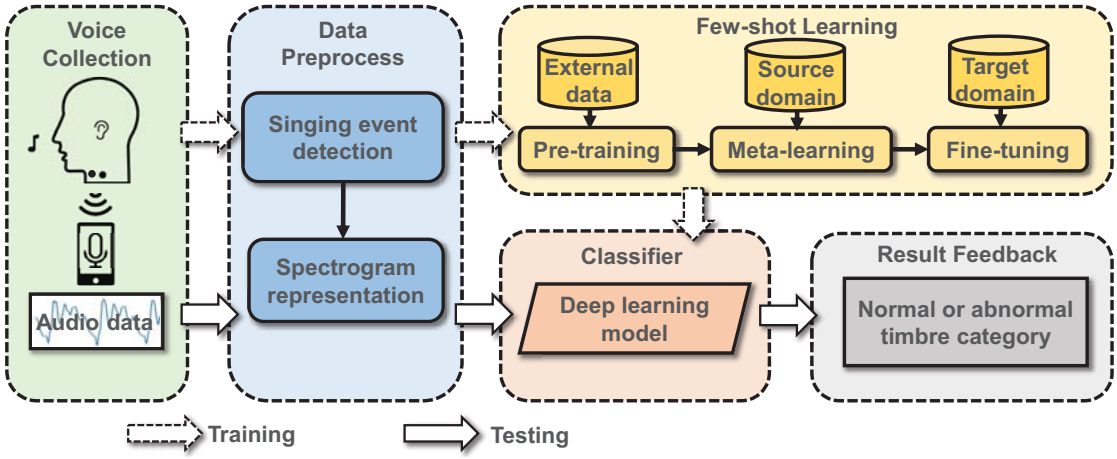


Fig. 1. The overview of TimbreSense.

instance, J. Sundberg et al. examined how phonation mechanisms of structures like the nasal cavity and vocal tract affect timbre characteristics [11, 21]. They also explored the connection between singing emotions and both physiological and acoustic indicators [33]. However, evaluating timbre solely based on acoustic features has limitations. This method is one sided and lacks clarity, as complex timbre cannot be fully captured by a few simple features. Additionally, the criteria for defining timbre through these features are vague, often relying on experts' subjective evaluations. Thus, this approach serves only as an auxiliary tool for timbre evaluation.

2.2 Machine Learning-based Singing Evaluation

With advancements in computing, machine learning and deep learning have been applied to acoustic feature analysis, primarily categorized into automatic singing evaluation and classification. Automatic singing evaluation uses extracted audio features as input and employs expert ratings as target outputs for the model [32, 34, 43]. Expert scoring incorporates subjective perceptual factors related to timbre, like mellowness and fullness. This approach depends heavily on expert knowledge, reducing the model's objectivity. Automatic singing classification effectively categorizes well-defined timbres with clear perceptual definitions, making it more objective than expert ratings. In recent decades, many studies have used machine learning to classify vocal patterns in singing [4, 15], including singer identification [17], voice disease diagnosis [20, 26], bel canto emotion recognition [35], and singing style classification [5, 28]. These studies do not address vocal abnormalities students frequently encounter during practice, nor have they been implemented in real-time systems. Our system evaluates five common timbre abnormalities that students frequently face during practice. This classification encompasses more categories than previous studies, adding to the challenge. We also introduce few-shot learning methods to enhance the performance of our model.

3 System Design

3.1 System Overview

This study presents TimbreSense, a timbre abnormality detection system specifically designed for bel canto. As shown in Figure 1, the system captures audio signals from smart devices. Singing event detection is used to extract relevant audio segments, converting them into spectrogram representations. The core architecture of our model is built around a transformer encoder. To boost performance across diverse user profiles, we implement few-shot learning during training,

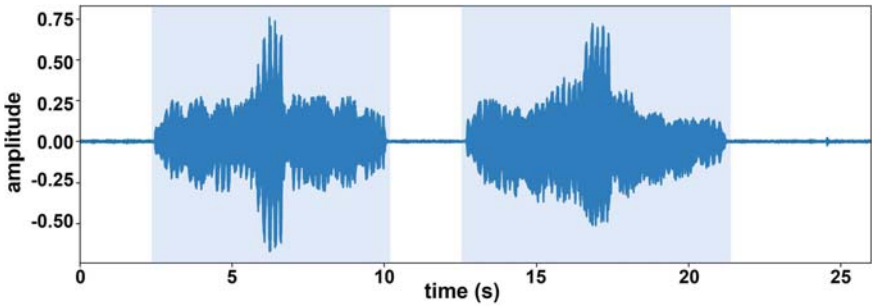


Fig. 2. Extracting valid audio segments.

treating each user profile as a distinct domain. Initially, the feature encoder is pre-trained on an open-source dataset, followed by fine-tuning using meta-learning techniques. In the testing phase, data from the target user are meticulously preprocessed and classified, providing precise indications of timbre abnormalities.

3.2 Data Preprocessing

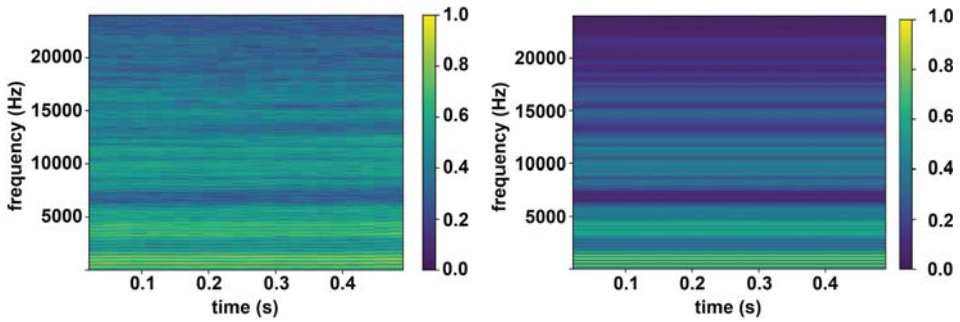
Detecting the singing voice is essential to minimize computing overhead from non-speech audio. We employ **Voice Activity Detection (VAD**) based on Gaussian Mixture Model from WebRTC [2]. The process's effectiveness is illustrated in Figure 2. After VAD extracts valid audio signals, pre-emphasis [40] is applied to enhance high-frequency components, resulting in a spectrogram with balanced frequencies. We then use non-overlapping 500-ms windows to segment the signal, transforming it into spectrograms via **Short-Time Fourier Transform (STFT)**. The audio is sampled at 48 kHz. Each segment length and FFT size are set to 2,048, with an overlap of 1,024 between segments.

3.3 Network Design

As stated in the Introduction, the correlation and semantic features of overtones in the frequency domain contain rich information, crucial for distinguishing different timbres. To extract meaningful timbral features, capturing effective global information from the spectrogram is essential. Transformers excel in extracting semantic features compared to RNNs and CNNs, demonstrating strong capabilities in capturing long-distance dependencies. Thus, we employ the transformer's self-attention mechanism to capture global information from the input data, effectively identifying long-range dependencies within the spectrogram. The attention mechanism is detailed in Equation (1), where Q, K, and V represent the query, key, and value matrices and d_k denotes the key vector dimension. Unlike CNNs and LSTMs, transformers use attention to enable parallel computation, enhance efficiency and better capture long-range dependencies. This approach enables effective capture of correlation and semantic features across different frequency bands in the spectrogram,

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \tag{1}$$

Our model, TimT, shown in Figure 4, is similar to ViT [7] but includes adaptive modifications to the input. In bel canto practice, vocal exercises typically involve slow variations in pitch across different tones. Over short durations, the loudness, pitch, and timbre of singing remain stable. Therefore, within short timeframes, the frequency of the audio signal can be assumed to remain constant. As shown in Figure 3(a), the energy of each frequency within the window remains relatively stable, with minor fluctuations at the boundaries of each peak. Given this, we apply temporal



(a) Original (b) Temporal average pooling

Fig. 3. Temporal average pooling on the STFT spectrogram: (a) Original spectrogram and (b) spectrogram after temporal average pooling.

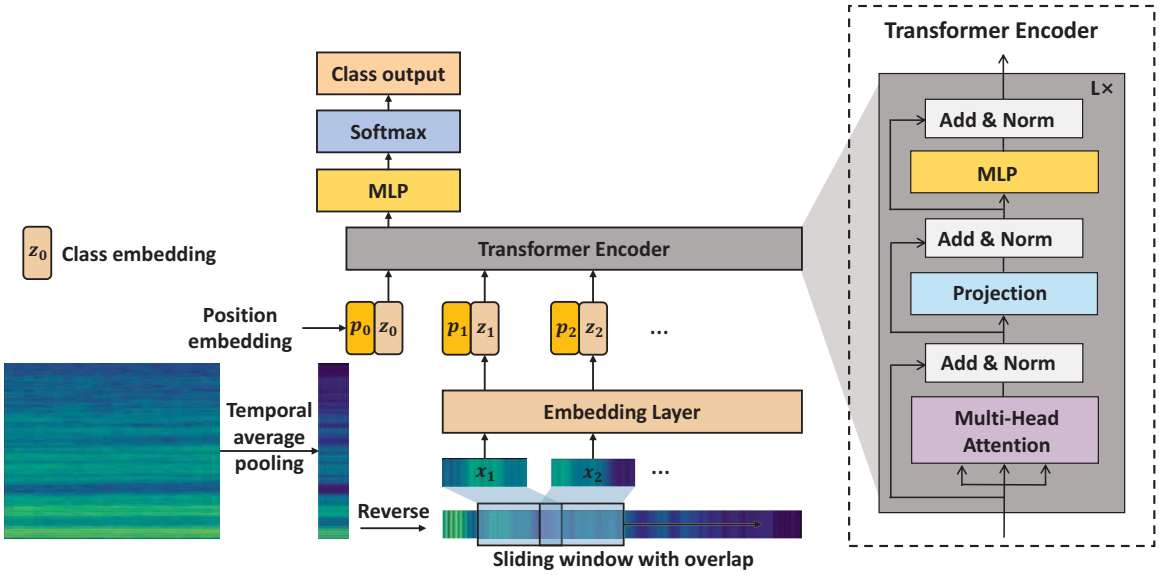


Fig. 4. The structure of TimT.

average pooling to the spectrograms, which smooths out inherent fluctuations and random noise, as shown in Figure 3(b). This process reduces temporal influence, directing the model's attention to frequency domain information. Next, we normalize the spectrogram and use a sliding window to divide the one-dimensional spectrogram $X \in \mathbb{R}^{F\times 1\times 1}$ into N fixed-size tokens, where F denotes the number of frequency bins. Each token x_i $(i = 1, 2, \ldots, N)$ passes through an embedding layer to generate its feature representation z_i. It is worth noting that, unlike ViT, we use overlapping windows to preserve some mutual information between tokens. In ViT, segmented images maintain edge continuity, while temporal-averaged spectrograms lack frequency direction continuity, complicating positional embedding learning. The token size is set to 64. Each token's feature representation z_i is added with a randomly initialized positional embedding p_i. Preserving mutual information between tokens aids in learning positional encoding during training. The class embedding z_0 is randomly initialized, with its dimension matching the feature dimension of the other token embeddings z_i. The class embedding is concatenated with the other token embeddings, passed through the positional embedding layer, and jointly fed into the transformer encoder. The class embedding engages in the self-attention calculations of the transformer and serves as the

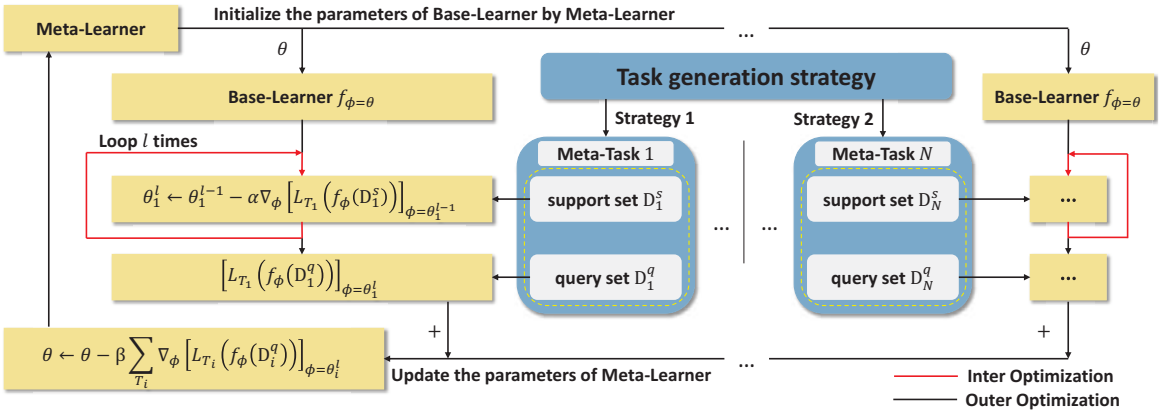


Fig. 5. The meta-learning process of TimbreSense.

final output, representing globally extracted spectral features. This feature then passes through an **Multi-Layer Perceptron (MLP)** followed by a Softmax layer to produce the class output for the sample.

The transformer encoder is composed of L layers ($L = 4$) of identical structure stacked sequentially. The encoder comprises multi-head attention, projection, and MLP modules. Residual connections and layer normalization are applied after each module. The multi-head attention module utilizes multiple sets of learnable queries, keys, and values, allowing the model to capture diverse internal relationships within the data. The projection block is a fully connected layer with zero bias, designed to reduce the feature dimension. The MLP consists of two layers equipped with GELU non-linearity.

3.4 Few-shot Learning for Cross-domain Recognition

Variations in vocal anatomy and singing techniques among individuals cause acoustically similar timbres to exhibit significant differences in their acoustic properties. This variability presents challenges to robust model generalization across different users. To tackle this and enhance cross-user capability, we adopt a few-shot learning approach, balancing user-friendliness and model performance. During the training phase, we leverage a comprehensive combination of pre-training, meta-learning, and fine-tuning techniques to effectively mitigate the cross-domain challenges arising from the heterogeneous acoustic attributes of different vocalists.

Pre-training. The transformer network inherently lacks inductive biases, making it necessary to acquire this prior knowledge through pre-training. We use the Vocalset [39] singing technique dataset for supervised pre-training of our model. Cross-entropy loss function is used to calculate the loss. We pretrain the model until the model exhibits satisfactory performance and the loss stabilizes. The encoder is then retained and connected to an untrained Softmax layer, whose output size matches the number of categories. This combined structure serves as the trainable model for the subsequent steps.

Meta-learning. **Model-Agnostic Meta-Learning (MAML)** [8] is a meta-learning algorithm specifically designed for few-shot learning. The goal of MAML is to train a model that can quickly adapt to new tasks with only a small amount of training data. MAML's core idea is to find an initial set of parameters θ through meta-learning, allowing the model to achieve good performance on new tasks after only a few gradient updates. The training process of *MAML*, illustrated in Figure 5, involves meta-tasks, a meta-learner, and base-learners. Meta-tasks are generated by our task generation strategy and are composed of small subsets of data from the source domain. Base-learners

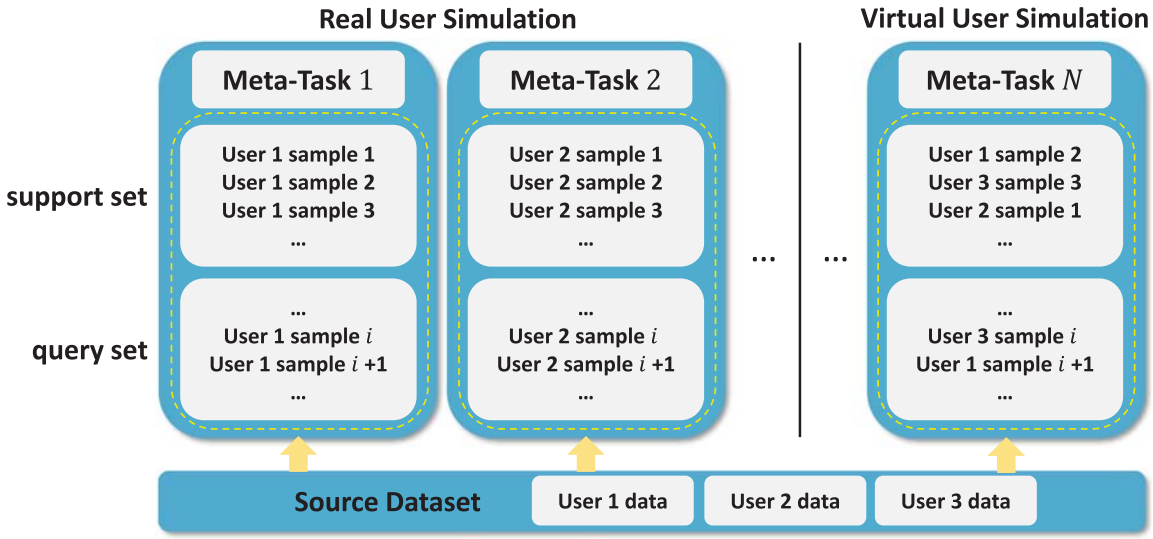


Fig. 6. Task generation strategy.

learn task-specific features on each meta-task, constructing target functions tailored to each task. The meta-learner integrates validation losses from multiple meta-tasks, identifying common patterns that facilitate generalization and enable rapid convergence on new tasks.

The composition of meta-tasks plays a critical role in the effectiveness of meta-learning. To optimize *MAML*, we enhanced the task generation strategy by introducing two approaches: **Real User Simulation (RUS)** and **Virtual User Simulation (VUS)**. As shown in Figure 6, assuming three users in the source domain data, in RUS, each meta-task's data are sampled exclusively from a single user's data. In VUS, each meta-task's data are randomly sampled across all users. For instance, in RUS, data for meta-task 1 come solely from User1, whereas, in VUS, data for meta-task N are randomly sampled from User1, User2, and User3. RUS aims to simulate real-world scenarios where the model learns from new users. VUS is designed to enhance user diversity, especially when training data from users are limited. We balance the number of tasks generated by RUS and VUS in each training batch. As depicted in Figure 6, we generate N meta-tasks, with the first $N/2$ tasks generated using the RUS strategy and the rest using VUS. This balanced approach integrates both strategies, enhancing the effectiveness of meta-learning. In our experiment, the number of tasks generated by RUS corresponds with the number of users in the training set. We collect data from 18 users, generating 810 meta-tasks through the RUS strategy (18 users $\times$ 45 tasks per user). The number of meta-tasks generated using the VUS strategy matches that of the RUS strategy. Therefore, the total number of meta-tasks (N) amounts to 1,620. Each meta-task is divided into a support set and a query set. The support set assists the model in learning task-specific patterns, while the query set evaluates the model's generalization to new domains with limited support data. The support set and query set configurations are 6-ways, 5-shot, and 6-ways, 10-shot, respectively. Here, "N-ways" denotes the number of categories in a task, while 'k-shots' represents the number of samples per category.

As shown in Figure 5, in the training process of *MAML*, the parameters of the base learner f_ϕ are first initialized by the parameters of meta-learner θ. In the inner optimization, the base-learner iterates over the support set D_i^s of the meta-task T_i for l iterations, and update the parameters:

$$\theta_i^l \leftarrow \theta_i^{l-1} - \alpha \nabla_\phi [\mathcal{L}_{T_i}(f_\phi(D_i^s))]_{(\phi=\theta_i^{l-1})}. \tag{2}$$

Once the base learners of all N meta-tasks have completed their iterations, the query sets D_i^q of each meta-task are used as a validation set. The current losses of each base learner are computed as follows:

$$[\mathcal{L}_{T_i}(f_\phi(D_i^q))]_{(\phi=\theta_i^l)} = \sum_j^n -y_{ji}^q \log\left(f_\phi(x_{ji}^q)\right) + (1-y_{ji}^q)\log\left(1-f_\phi(x_{ji}^q)\right), \tag{3}$$

where y_{ji}^q and x_{ji}^q, $j \in (1, 2, \ldots, n)$, are the samples of the query set D_i^q. Finally, in the outer optimization, the parameters of the meta-learner are updated through

$$\theta \leftarrow \theta - \beta \sum_{T_i} \nabla_\phi [\mathcal{L}_{T_i}(f_\phi(D_i^q))]_{\phi=\theta_i^l}. \tag{4}$$

We use the cross-entropy loss function along and the Adam optimization algorithm with a weight decay set to 0.0001 to compute losses and update gradients. The paramaters α and β represent the learning rates for inner and outer optimization, respectively. Specifically, α is set to 0.002, and β is set to 0.00015 empirically. The number of iterations for inner optimization l is 5, and for outer optimization it is 15.

Fine-tuning. The final stage of few-shot learning is fine-tuning. After meta-learning, the model needs only a few samples from target users to achieve domain generalization. The tasks encountered by the model during this stage are consistent with those generated using the RUS strategy. Therefore, we use the same learning rates for the fine-tuning stage as mentioned earlier. We freeze the parameters of the transformer encoder and fine-tune only the subsequent classification head. We set the step size S empirically. When the number of iterations reaches S, we reduce the learning rate to facilitate stable convergence of the model towards the optimal position,

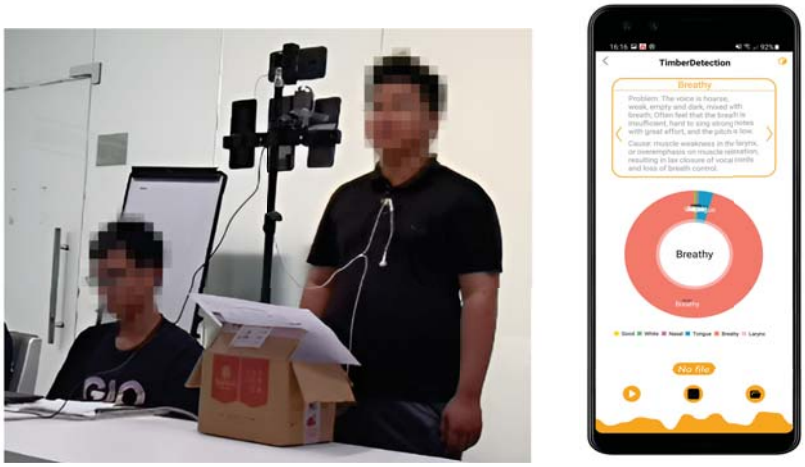
$$\theta^l = \begin{cases} \theta^{l-1} - \alpha \nabla_\phi [\mathcal{L}(f_\phi)]_{(\phi=\theta^{l-1})}, & 1 \le l \le S \\ \theta^{l-1} - \alpha_s \nabla_\phi [\mathcal{L}(f_\phi)]_{(\phi=\theta^{l-1})}, & S < l \end{cases}, \tag{5}$$

where $\alpha = 0.002$, $\alpha_s = 0.001$, $S = 10$.

4 Implementation

4.1 Data Collection

We recruited 18 bel canto students (9 males and 9 females). Each student, guided by an experienced vocal coach, imitated sounds with timbre abnormalities. As shown in Figure 7(a), the experiments were conducted in an uncontrolled meeting room with background noise of approximately 40 dB. Students were asked to perform vocal exercises from the standard bel canto training repertoire. They were required to sing the five foundational bel canto vowels (/a/, /e/, /i/, /o/, /u/) at various pitches. Pre-recorded audio segments of abnormal timbres were played, guiding students via an app to imitate them. Students maintained these timbres while performing vocal exercises. A vocal coach was actively involved throughout the entire data collection process. After the experiment concluded, the coach provided guidance on correct vocal techniques to protect students' vocal health. Each timbre abnormality has 15 to 20 samples, with each valid segment ranging from 7 to 13 seconds. After collecting 5 samples, we ask the participants about their condition. If they are in good condition, then we continue with the collection; otherwise, they rest for ~1 to 2 minutes. As shown in Table 1, four smartphones were used, RedMi K30s, Vivo X20, Samsung S8, and Samsung S20, to record the participants' singing. Except for the Samsung S8, which used a wired headphone microphone with a 3.5-mm audio interface, the other smartphones used their built-in microphones. The smartphones were vertically mounted on stands at the same height as the participant's head, approximately 30 cm from the mouth. The headphone microphone was clipped at the participant's collar After data collection, a professional vocal coach was responsible for cleaning the timbre



(a) Bel canto audio data collection (b) TimbreSense APP

Fig. 7. Data collection and system deployment: (a) Bel canto audio data collection and (b) TimbreSense APP.

Table 1. Device Parameters

Device	Sensor	CPU	RAM
RedMi K30s	Built-in microphone	2.8 GHz	8 G
Vivo X20	Built-in microphone	2.2 GHz	4 G
Samsung S20	Built-in microphone	2.8 GHz	12 G
Samsung S8	Wired headphone microphone	2.3 GHz	4 G

data. The coach screened out samples not aligning with the target abnormality category, discarding about 15% of them. In total, 1,944 timbre samples are collected across all categories for each device, and after data cleaning, 1,640 samples were retained.

5 Evaluation

5.1 System Implementation

We develop an app for our system to facilitate audio recording and model inference, as shown in Figure 7(b). Audio is recorded at a sampling rate of 48 KHz and saved as WAV files. The deep learning model in our system is built on PyTorch-1.11.0 and deployed on mobile devices. All training processes are conducted on the server, while the mobile device handles the model inference. We developed an app on the smartphone platform for our system. The app first performs VAD, followed by processing of the detected valid audio segments. Finally, it performs model inference on the mobile device and returns the timbre abnormality results.

5.2 Experiment Setup

We employ two distinct data partitioning approaches for our experiments. For the Backbone Network Experiments, we conduct 10-fold cross-validation on data from all users without utilizing few-shot learning. This approach assesses the performance of the backbone network in a non-cross-user scenario, establishing an upper limit for the model's performance in cross-user tasks. In the remaining tests, we employ the leave-one-user-out method to partition training and testing data, evaluating the performance of few-shot learning methods in cross-user scenarios. We assess the accuracy of the models and the standard deviation across multiple test sets.

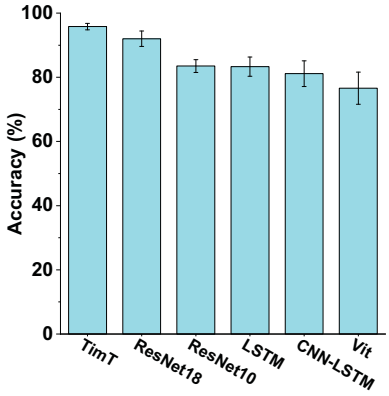


Fig. 8. Backbone networks comparison of 2D-input.

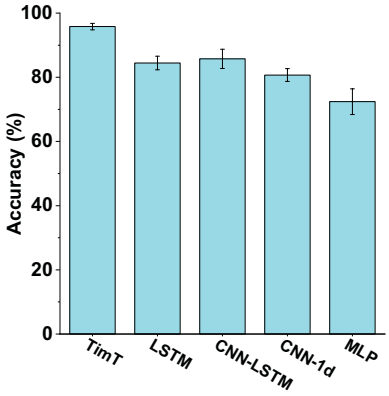


Fig. 9. Backbone networks comparison of 1D-input.

5.3 Selection of Backbone Network

In the Backbone Network Experiments, we evaluate the performance of models with original spectrogram inputs (2D) and models with temporal average-pooled spectrogram inputs (1D). We also assess the impact of window overlap rates on the token segmentation in our model.

5.3.1 Original Spectrogram Inputs (2D). We compare the performance of conventional audio processing models to *TimT*. Conventional methods directly use the audio spectrogram obtained via STFT as input, differing from the temporal average pooling proposed in this article. We evaluate several classic models, including *ResNet18* [12], *ResNet10* [30], *LSTM* [13], *CNN-LSTM*, *MLP* (Multilayer Perceptron), and the original *ViT* [7]. *LSTM* and *CNN-LSTM* are commonly used for audio processing [5, 17, 35]. *ResNet18*, *ResNet10*, *ViT*, *LSTM*, and *CNN-LSTM* all use the classic 2D-spectrogram input, while our model use the temporal average pooling 1D-spectrogram as input to match its model structures. As Figure 8 shows, our model achieves an accuracy of 95.8%. Compared with common models like *LSTM* and *CNN-LSTM*, our model shows improvements of 12.5% and 14.7%, respectively. The accuracy of *ResNet18* approaches that of *TimT* at 92.0%. However, our model contains only 1.3 M parameters, significantly fewer than *ResNet18*'s 33 M. Additionally, we demonstrate the effectiveness of temporal average pooling. The original *ViT* model achieved an accuracy of only 76.6%. Temporal average pooling mitigates the influence of time, allowing the model to focus more on global correlations among overtones that form the timbre. Therefore, the model's recognition accuracy improves significantly.

5.3.2 Temporal Average-pooled spectrogram Inputs (1D). We evaluate the feature extraction capabilities of different models on 1D inputs and select common 1D processing models for comparison. The input size for *LSTM* and *CNN-LSTM* matches the token size of our model, both set at 64. As shown in Figure 9, our model achieves an accuracy of 95.8%, exceeding *LSTM* and *CNN-LSTM* by 11.3% and 10.0%, respectively, and surpassing *CNN-1d* and *MLP* by 15.1% and 23.4%, respectively. This demonstrates that the transformer model based on attention mechanisms effectively captures overtone features in the frequency domain and is well suited for our task.

5.3.3 The Impact of Overlap of Token Segment. We conduct experiments on the token segmentation window overlap rate, demonstrating the effectiveness of retaining partial information between tokens. As shown in Figure 11, the model's recognition accuracy increases with the increase in overlap rate, reaching its peak at 50%, and then decreases. When the overlap rate becomes too high, excessive redundancy between adjacent frames causes the model to encounter

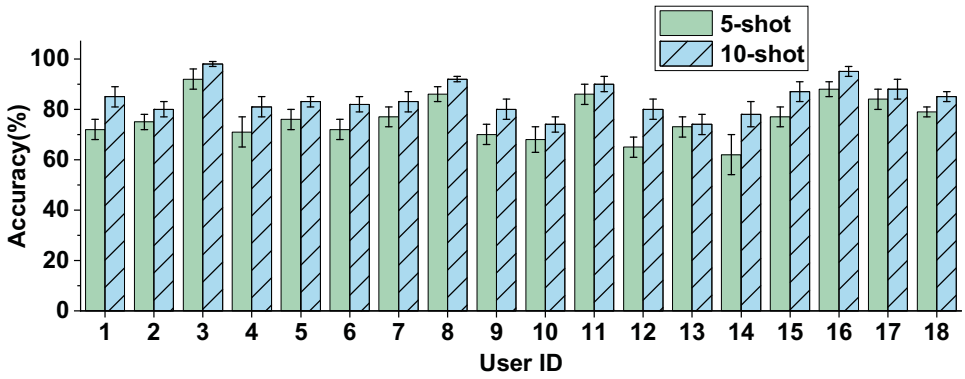


Fig. 10. The cross-domain performance of the system.

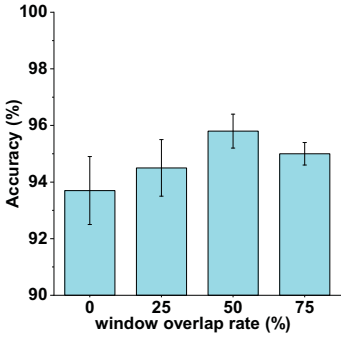
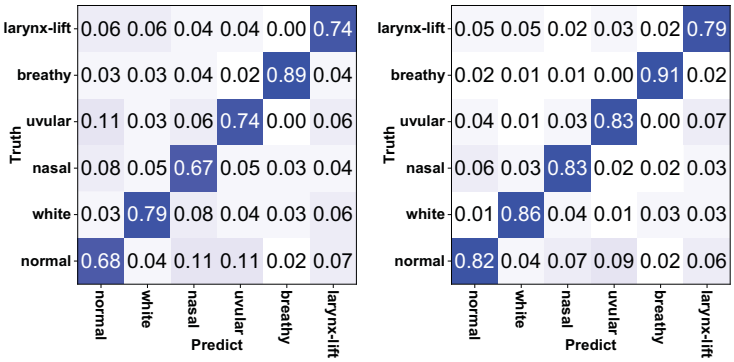


Fig. 11. Window overlap rate comparison.



(a) 5-shot (b) 10-shot

Fig. 12. Confusion matrix of different timbre types: (a) 5-shot scenario; (b) 10-shot scenario.

similar information during training, resulting in overfitting. Thus, considering accuracy, we select a 50% overlap rate as the optimal setting.

5.4 Overall Performance

We employ the leave-one-user-out method for data partitioning and train the data using our proposed model and few-shot learning approach. Figure 10 shows the model's performance under different new user scenarios, with evaluations conducted under 5-shot and 10-shot settings. For 5-shot, the average accuracy is 76.3%, with User3 achieving the highest accuracy of 92.6% and User14 the lowest at 61.9%. For 10-shot, the average accuracy improves to 84.5%. Comparing 5-shot and 10-shot, the 10-shot version shows varying degrees of accuracy improvement for all users, demonstrating more stable performance.

To understand the misclassification of the system, we construct a classification confusion matrix shown in Figure 12. Compared to other categories, *white* and *breathy* show the fewest misclassifications, while *normal* and *nasal* have the highest misclassification rates. This may be due to the fact that, aside from *normal* and *nasal*, other categories of abnormal timbres possess more distinct acoustic characteristics, making them easier for the model to differentiate. *Normal* is defined as the sound produced by the user that conforms to the norms of classical singing, and its auditory characteristics are primarily reflective of the user's unique timbre. In contrast, other categories of abnormal timbres share common, interpretable auditory features. For instance, *breathy* sounds

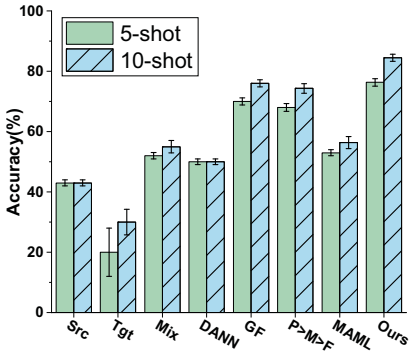


Fig. 13. Baseline comparison of few-shot learning.

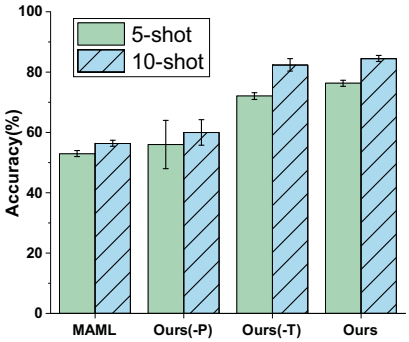


Fig. 14. Ablation experiment.

have a breathy quality, while *uvular* sounds are perceived as stiff and dark. Consequently, the model faces more challenging to capture the characteristics of *normal* phonation.

5.5 The Comparison with Baseline Methods

We employ the following methods as baselines to assess the cross-domain performance of our system:

- *Src*: The model is trained solely on source domain data without further adjustments using target domain data.
- *Tgt*: Limited samples are extracted from the target domain for training, and the remaining data are used for testing.
- *Mix*: The model is trained on a combination of limited target domain samples and all source domain data, with the remaining target domain data used for testing.
- **Domain-Adversarial training of Neural Networks (DANN)** [9]: Domain-Adversarial training of Neural Networks is a prominent domain adaptation approach requiring substantial unlabeled target domain data for adaptation.
- *GF*: Global fine-tuning was initially trained using data from the source domain and further fine-tuned with limited samples from the target domain.
- *P>M>F* [14]: This is a state-of-the-art few-shot learning approach following a pre-training, meta-learning, fine-tuning pipeline similar to our method. The key difference lies in its core meta-learning algorithm, which is the Prototypical Network [31]. Prototypical Network represents a metric-based meta-learning approach that generates prototypes for each class using limited labeled data from target users. Recognition results are determined by comparing distances between test samples and prototypes during testing.
- *MAML* [8]: Model-Agnostic Meta-Learning is a widely used and significant optimization-based meta-learning method. In our evaluation, this refers specifically to the original, unmodified *MAML* method. Our approach differs in the task generation strategy compared to the original *MAML* method.

We use the same backbone network, evaluating all methods under 5-shot and 10-shot conditions. A leave-one-user-out evaluation strategy is used. Among baseline methods, *Src* uses only source domain data. *DANN* utilizes both source domain data and approximately 400–600 unlabeled target domain samples. Thus, these two baselines are not dependent on shot count.

As shown in Figure 13 and Table 2, we compare the results of baseline methods. *Src* achieves an accuracy of only 43.3%, significantly lower than the 95.8% shown in Figure 8 under non-cross-user

Table 2. Baseline Comparison of Few-shot Learning

Accuracy	Src	Tgt	Mix	DANN	GF	P>M>F	MAML	Ours
5-shot	43.1(±1.3)	20.1(±8.5)	52.3(±1.1)	50.6(±1.5)	70.4(±1.2)	68.7(±1.3)	53.5(±1.4)	76.3(±1.2)
10-shot	43.1(±1.3)	30.6(±4.2)	55.8(±2.1)	50.6(±1.5)	76.9(±1.2)	74.3(±1.6)	56.4(±2.2)	84.5(±1.2)

conditions. This demonstrates that for the same abnormal timbre categories among different individuals, strong individual variability persists, resulting in poor cross-domain performance. *Tgt*, trained with very few target domain samples without cross-domain training, shows the lowest performance and suffers from severe overfitting. *DANN* utilizes substantial unlabeled target domain data, but its performance fails to meet basic requirements, indicating the unsuitability of unsupervised domain adaptation methods for bel canto timbre abnormal detection. *GF* is trained with source domain data and fine-tuned with a small amount of target domain data, achieving relatively good performance but still lagging behind our method. *P>M>F*, a prototype-based meta-learning method, performs worse than *GF*, suggesting the limitations of prototype-based methods in handling features with significant individual differences. The original *MAML* performs poorly due to the lack of pre-training. Its transformer encoder lacks prior inductive biases, contributing to inferior performance. Overall, our method achieves the best recognition performance in both 5-shot and 10-shot conditions, demonstrating that our proposed few-shot cross-domain approach, integrating pre-training, meta-learning, and fine-tuning, enhances cross-domain performance, and outperforms other methods.

5.6 Ablation Experiment

As shown in Figure 14, we conduct ablation experiments for our few-shot cross-domain strategy to demonstrate its effectiveness. We evaluate the effects of removing the pre-training step (-P) and the user domain simulation strategy in meta-learning task generation (-T). All methods undergo testing under 5-shot and 10-shot conditions using leave-one-user-out cross-validation. *Ours(-P)* performs similarly to *MAML* due to the absence of external dataset pre-training, yet shows noticeable improvement. This suggests the importance of pre-training to introduce prior knowledge into the transformer. *Ours(-T)* demonstrates that the user-domain simulation strategy can enhance the model's cross-domain recognition performance.

5.7 The Impact of Number of Shots

In the fine-tuning phase, we provide varying samples to evaluate the impact of different target domain sample quantities on the model's performance. As shown in Figure 15, when the sample number ranges from 1-shot to 10-shot, the model's accuracy rapidly improves as shot counts increase. A clear slowdown in accuracy gains is observed between 5-shot and 10-shot. Beyond 10-shot, accuracy saturates. Considering user costs and recognition accuracy, we select 5-shot as the final input quantity during fine-tuning.

It is worth noting that the *shot* provided during few-shot learning is different from the original audio sample users provide during data collection. The shots that used for fine-tuning to enhance accuracy are 0.5-second segments from window splits during data preprocessing. An audio clip can be divided into multiple shots, reducing the required amount of user-provided audio data. In practice, users only need to provide about 5 seconds of data per category to achieve the enhancement effect of 10 shots. The method by which we obtain new user-labeled samples is consistent with the data collection approach outlined in Section 4.1. The mobile app contains pre-recorded abnormal timbre samples. When obtaining new user-labeled samples, the new user plays back an audio sample of a certain category of abnormal timbre and, under the guidance of the app, imitates

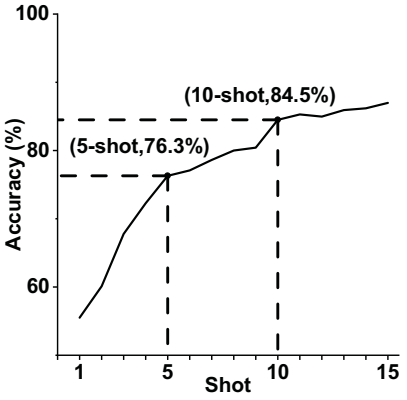


Fig. 15. Impact of fine-tuning sample size.

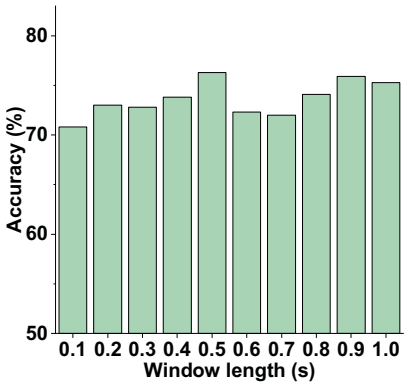


Fig. 16. Impact of window length on accuracy.

the abnormal timbre to produce a labeled sample. This approach efficiently collects labeled data from new users with minimal effort.

5.8 The Impact of Detecting Window Length

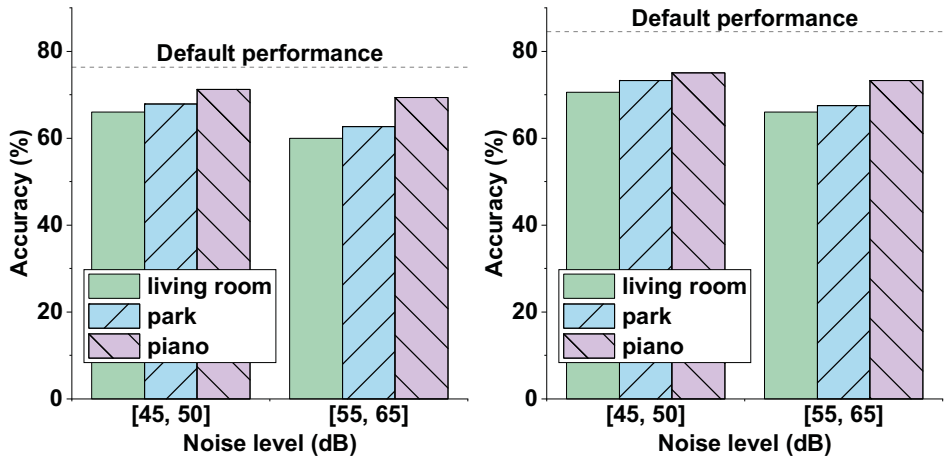
We evaluate the impact of varying input data window lengths on the model's recognition performance. As shown in Figure 16, we conduct leave-one-out cross-validation experiments to analyze the effect of varying window lengths under the 5-shot condition. The model's recognition accuracy increase with the increase in window length, reaching its peak at 0.5 seconds. Considering both real-time usage and recognition performance, we ultimately chose a window length of 0.5 seconds.

5.9 The Impact of Noise

To assess the system's robustness to environmental factors, we perform noise impact experiments. We select three common types of background noise: piano accompaniment, living room environmental noise, and public environmental noise. Environmental noise is sourced from the public DEMAND dataset [36]. The noise volume is controlled within ranges of [45–50] and [55–65] decibels and is linearly combined with the user's vocal data to simulate a noisy voice signal. The experimental results are shown in Figure 17, where the baseline represents performance without added noise. Among the three different types of noise, the model performs better with piano accompaniment compared to environmental noise. This is because environmental noise is typically wide-band random noise, which unpredictably distorts the spectral characteristics of the user's singing. In contrast, piano accompaniment exists in a narrow frequency range and has a smaller impact on the spectral characteristics of the user's singing. Additionally, as noise levels increase, the model's recognition performance slightly decreases, though users can enhance noise resilience by providing more data. In summary, the model is suitable for relatively quiet environments and performs well in scenarios with piano accompaniments, making it applicable for bel canto singing practice.

5.10 The Impact of Loudness

Given that the average loudness of all audio samples in the dataset is approximately −15 LUFS, we evaluate the system's performance at loudness levels of −5, −10, −15, −20, and −25 LUFS for both 5-shot and 10-shot scenarios. Loudness-adjusted audio is used for both fine-tuning and testing, as the loudness of the same user's pronunciation during practice in real scenarios is usually consistent. The results show that accuracy remains consistent across different loudness levels, with



(a) 5-shot (b) 10-shot

Fig. 17. Evaluation of the robustness to noise: (a) 5-shot scenario; (b) 10-shot scenario.

average accuracies of 76.3% for 5-shot and 84.5% for 10-shot. This occurs because adjusting the audio loudness essentially scales the amplitude of the signal without changing the inherent signal-to-noise ratio of the signal. We assume the new audio signal $y(t)$ is a scaled version of the original $x(t)$, multiplied by a constant factor α. The STFT has a linear property, meaning that if a signal $x(t)$ is multiplied by a constant factor α, then its Fourier transform $X(f)$ is also multiplied by the same factor α. During data preprocessing in our system, the spectrograms after global temporal pooling are normalized, eliminating the influence of the constant factor α. As a result, even if the audio loudness changes, the normalized spectrograms remain consistent, ensuring that the model's input does not vary with loudness. This demonstrates the system's robustness to variations in loudness.

5.11 Running Performance on Mobile Devices

Real-time singing detection is of importance as it allows users to promptly assess the quality of their vocal performance. Consequently, the system's response time becomes a critical factor impacting the overall user experience. To thoroughly evaluate the system's response time, we divide the processing flow into four distinct stages:

- —Model loading: loading the deep learning model into memory to create a runnable classifier. This step is only executed once, and the model's parameter size directly affects loading speed.
- —Signal processing: pre-processing of the input raw audio stream, including signal processing operations such as singing event detection and time-frequency feature extraction.
- —Model inference: the deep learning classifier processes input spectrogram features to produce prediction results.
- —Result presentation: the time for model prediction results to be converted into visualized user feedback.

A series of experiments measure the response time of real-time singing voice input on four devices. To ensure accuracy, multiple trials are conducted, with each trial involving opening the main interface, waiting for approximately 3 seconds after activating the recording button, singing for around 7 seconds, and finally stopping the recording and exiting the main interface. This process is repeated 10 times, and the average response time is calculated. As shown in Figure 18,

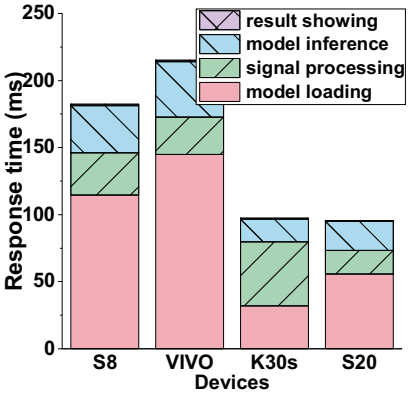


Fig. 18. Response time.

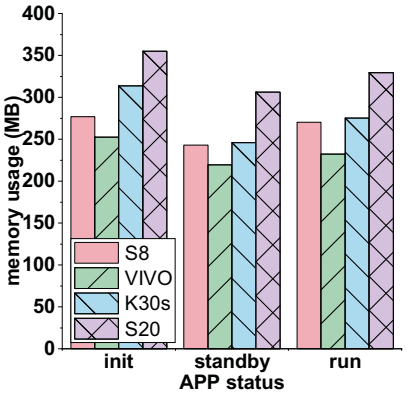


Fig. 19. Memory usage.

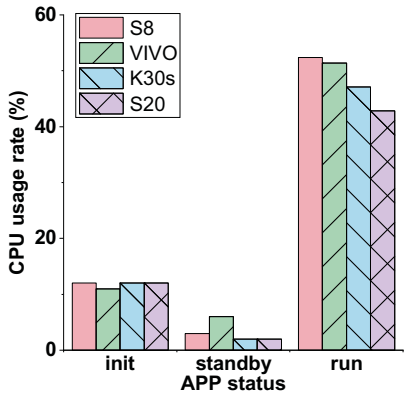


Fig. 20. CPU usage.

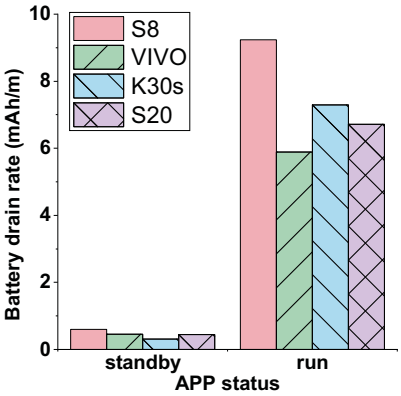


Fig. 21. Battery drain rate.

the response time for the model on mobile devices consists of several stages. The maximum response time for all devices is 215.2 ms, while the minimum is 95.6 ms. The average response time is 147.6 ms. Samsung S20 and RedMi K30s perform better due to their higher CPU frequencies. Overall, the model deployed on mobile devices, from loading to outputting the audio abnormal timbre result, requires less than 250 ms, which meets the real-time requirements for user usage.

We also evaluate the power consumption, CPU usage, and memory usage at different stages of the app's operation. We conduct evaluations during several app operation stages, including the initialization (init) phase, standby phase, and model inference (run) phase. During the initialization phase, the device initializes the environment and loads the local model. In the standby phase, the app activates the recording function, but the user is not singing. During this phase, the app executes the VAD algorithm to detect singing events, and if no singing is detected, then it does not perform model inference. In the model inference phase, the app records while the user continues singing. At this point, the VAD algorithm detects the singing event, initiates data processing, performs model inference, and ultimately returns the abnormal timbre result. As shown in Figure 19, memory usage across all devices stays under 400 MB in all phases. As shown in Figure 20 and Figure 21, the system's CPU usage during the standby phase is at most 6%, and the maximum power consumption rate is only 0.595 mAh/minute. This occurs because, in standby mode, singing event detection removes invalid audio, avoiding unnecessary data processing and inference. In the model inference phase, the model's CPU usage reaches a maximum of 52.4%, and the maximum power consumption rate is 9.24 mAh/minute.

6 Discussion

This article proposes a system for detecting timbre abnormalities in bel canto singing. However, the system has limitations. First, user diversity presents a challenge. In cross-user recognition, significant differences in vocal organs lead to a notable drop in model performance (from 95.8% in non-cross-user scenarios to 43.1% in cross-user *Src* scenarios). To address this, we propose that users imitate the atypical vocal tones generated by the app, providing the model with a few samples for few-shot learning to improve performance. In the future, we will reduce the cost for users by minimizing the required sample size. Second, this work focuses on timbre abnormalities during student practice sessions. However, other vocal issues, such as inaccurate pitch and lack of emotional expression, arise during practice. Pitch representation on a spectrogram is straightforward, as different pitches correspond to distinct fundamental frequencies and harmonic distributions, remaining consistent across individuals. Pitch evaluation in bel canto practice can be performed using Fourier transforms or deep learning models. Vocal emotions are primarily conveyed through tone, style, timbre, changes in strength, and rhythm [35]. These features can be effectively captured in audio spectrograms. In Reference [35], spectrograms and various low-level machine learning features serve as audio features, while a deep learning model classifies musical emotions. This work utilizes spectrograms of vocal audio as features for classifying abnormal timbres, which also include other vocal characteristics such as pitch, changes in strength, and style. Thus, additional practice content can be integrated based on these audio features, creating a comprehensive training system for bel canto. In the future, we will continue to enhance the system's capabilities.

7 Conclusion

This article presents a timbre abnormality detection system specifically designed for trainees in bel canto practice. Our system achieves real-time performance and high generalization capabilities by incorporating temporal average pooling and a transformer encoder to capture essential correlations and semantic features of overtones. Furthermore, we apply a few-shot learning approach, which combines pre-training, meta-learning, and fine-tuning, to enhance the system's generalization performance. Experimental results demonstrate promising cross-user recognition accuracies of 76.3% and 84.5% under 5-shot and 10-shot conditions, respectively. Remarkably, the average system response latency is 147.6 ms, satisfying practical usage requirements. Moving forward, our future endeavors will focus on further improving the system's recognition performance and noise robustness, as well as exploring additional application scenarios for its utilization.

References

[1] Gerrit Bloothooft, Eldrid Bringmann, Marieke Van Cappellen, Jolanda B. Van Luipen, and Koen P. Thomassen. 1992. Acoustics and perception of overtone singing. *J. Acoust. Soc. Am.* 92, 4 (1992), 1827–1836.

[2] Niklas Blum, Serge Lachapelle, and Harald Alvestrand. 2021. WebRTC: Real-time communication for the open web platform. *Commun. ACM* 64, 8 (2021), 50–54.

[3] Pasquale Bottalico, Mark T. Marunick, Charles J. Nudelman, Jossemia Webster, and Maria Cristina Jackson-Menaldi. 2021. Singing voice quality: The effects of maxillary dental arch and singing style. *J. Voice* 35, 3 (2021), 501.e11–501.e18.

[4] Manuel Brandner, Paul Armin Bereuter, Sudarsana Reddy Kadiri, and Alois Sontacchi. 2023. Classification of phonation modes in classical singing using modulation power spectral features. *IEEE Access* 11 (2023), 29149–29161.

[5] Qiao Chen, Wenfeng Zhao, Qin Wang, and Yawen Zhao. 2022. The sustainable development of intangible cultural heritage with AI: Cantonese opera singing genre classification based on CoGCNet model in China. *Sustainability* 14, 5 (2022), 2923.

[6] Matthew Derek Cyphert. 2022. *The Most Common Vocal Fault in the Baritone Voice.* . West Virginia University.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv:2010.11929*. Retrieved from https://arxiv.org/abs.2010.11929

[8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*. PMLR, 1126–1135.
[9] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* 17, 1 (2016), 2096–2030.
[10] Chitralekha Gupta, Jinhu Li, and Haizhou Li. 2021. Towards reference-independent rhythm assessment of solo singing. In *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC'21)*. IEEE, 912–919.
[11] Miriam Havel, Tanja Kornes, Eddie Weitzberg, Jon O. Lundberg, and Johan Sundberg. 2016. Eliminating paranasal sinus resonance and its effects on acoustic properties of the nasal tract. *Logopedics Phoniatrics Vocology* 41, 1 (2016), 33–40.
[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.
[13] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* 9, 8 (1997), 1735–1780.
[14] Shell Xu Hu, Da Li, Jan Stühmer, Minyoung Kim, and Timothy M. Hospedales. 2022. Pushing the limits of simple pipelines for few-shot learning: External data and fine-tuning make a difference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9068–9077.
[15] Sudarsana Reddy Kadiri, Paavo Alku, and Bayya Yegnanarayana. 2020. Analysis and classification of phonation types in speech and singing voice. *Speech Commun.* 118 (2020), 33–47.
[16] Elizabeth Johnson Knight and Stephen F. Austin. 2020. The effect of head flexion/extension on acoustic measures of singing voice quality. *J. Voice* 34, 6 (2020), 964.e11–964.e21.
[17] Seyed Kooshan, Hashemi Fard, and Rahil Mahdian Toroghi. 2019. Singer identification by vocal parts detection and singer classification using lstm neural networks. In *Proceedings of the IEEE International Conference on Pattern Recognition and Image Analysis (IPRIA'19)*. IEEE, 246–250.
[18] Pauline Larrouy-Maestri, David Magis, and Dominique Morsomme. 2014. Effects of melody and technique on acoustical and musical features of western operatic singing voices. *J. Voice* 28, 3 (2014), 332–340.
[19] Björn Lindblom and Johan Sundberg. 1971. Acoustical consequences of lip, tongue, jaw, and larynx movement. *The Journal of the Acoustical Society of America* 50, 4B (1971), 1166–1179.
[20] Mario Madruga, Yolanda Campos-Roca, and Carlos J Pérez. 2021. Impact of noise on the performance of automatic systems for vocal fold lesions detection. *Biocybernetics and Biomedical Engineering* 41, 3 (2021), 1039–1056.
[21] Alexander Mainka, Anton Poznyakovskiy, Ivan Platzek, Mario Fleischer, Johan Sundberg, and Dirk Mürbe. 2015. Lower vocal tract morphologic adjustments are relevant for voice timbre in singing. *PLoS One* 10, 7 (2015), e0132241.
[22] James C. McKinney. 2005. *The Diagnosis and Correction of Vocal Faults: A Manual for Teachers of Singing and for Choir Directors*. Waveland Press.
[23] Daryush D. Mehta, Jarrad H. Van Stan, Matías Zañartu, Marzyeh Ghassemi, John V Guttag, Víctor M Espinoza, Juan P Cortés, Harold A Cheyne, and Robert E Hillman. 2015. Using ambulatory voice monitoring to investigate common voice disorders: Research update. *Frontiers in Bioengineering and Biotechnology* 3 (2015), 155.
[24] Sarah Khatcherian Milo. 2014. *The Voice Teacher's Guide to Vocal Health for Voice Students: Preventing, Detecting, and Addressing Symptoms*. Ph.D. Dissertation. The Ohio State University.
[25] Koichi Omori, Ashutosh Kacker, Linda M. Carroll, William D. Riley, and Stanley M. Blaugrund. 1996. Singing power ratio: Quantitative evaluation of singing voice quality. *Journal of Voice* 10, 3 (1996), 228–235.
[26] Andrew J. Ortiz, Laura E. Toles, Katherine L. Marks, Silvia Capobianco, Daryush D. Mehta, Robert E. Hillman, and Jarrad H. Van Stan. 2019. Automatic speech and singing classification in ambulatory recordings for normal and disordered voices. *The Journal of the Acoustical Society of America* 146, 1 (2019), EL22–EL27.
[27] Polina Proutskova, Christophe Rhodes, Geraint Wiggins, and Tim Crawford. 2012. Breathy or resonant-a controlled and curated dataset for phonation mode detection in singing. In *13th International Society for Music Information Retrieval Conference*. 589–594.
[28] Jean-Luc Rouas and Leonidas Ioannidis. 2016. Automatic classification of phonation modes in singing voice: Towards singing style characterisation and application to ethnomusicological recordings. In *Interspeech*, Vol. 2016. 150–154.
[29] Gláucia Laís Salomão and Johan Sundberg. 2010. Perceptual relevance of voice source characteristics in male singers' modal and falsetto registers. In *Proceedings of the 5th International Conference on the Physiology & Acoustics of Singing*. 29–29.
[30] Marcel Simon, Erik Rodner, and Joachim Denzler. 2016. Imagenet pre-trained models with batch normalization. *arXiv preprint arXiv:1612.01452* (2016).
[31] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems* 30 (2017), 1–11.

[32] Xiaoheng Sun, Yuejie Gao, Hanyao Lin, and Huaping Liu. 2023. Tg-Critic: A timbre-guided model for reference-independent singing evaluation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'23)*. IEEE, 1–5.

[33] Johan Sundberg, Gláucia Laís Salomão, and Klaus R Scherer. 2024. Emotional expressivity in singing. assessing physiological and acoustic indicators of two opera singers' voice characteristics. *The Journal of the Acoustical Society of America* 155, 1 (2024), 18–28.

[34] Terry Tan. 2019. Singing evaluation based on deep metric learning. In *Proceedings of the 3rd International Symposium on Computer Science and Intelligent Control*. ACM, 1–5.

[35] Zhangcheng Tang. 2022. Music sense analysis of bel canto audio and bel canto teaching based on LSTM mixed model. *Mobile Information Systems* 2022, 1 (2022), 1875815.

[36] Joachim Thiemann, Nobutaka Ito, and Emmanuel Vincent. 2013. The diverse environments multi-channel acoustic noise database (demand): A database of multichannel environmental noise recordings. In *Proceedings of Meetings on Acoustics*, Vol. 19. AIP Publishing.

[37] Cheruvathur Uthup. 2016. *The Acoustical Foundations of Bel Canto*. Ph.D. Dissertation. Indiana University.

[38] Cora-Mari Van Vuuren et al. 2017. *Exploring the Diagnosis and Correction of Vocal Faults Encountered During the Training of the Classical Singing Voice*. Ph.D. Dissertation. University of Pretoria.

[39] Julia Wilkins, Prem Seetharaman, Alison Wahl, and Bryan Pardo. 2018. VocalSet: A singing voice dataset. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'18)*. 468–474.

[40] D Wong, C Hsiao, and J Markel. 1980. Spectral mismatch due to preemphasis in LPC analysis/synthesis. *IEEE Trans. Acoust. Speech Sign. Process.* 28, 2 (1980), 263–264.

[41] Fusang Zhang, Jie Xiong, Zhaoxin Chang, Junqi Ma, and Daqing Zhang. 2022. Mobi2Sense: Empowering wireless sensing with mobility. In *Proceedings of the 28th Annual International Conference on Mobile Computing and Networking*. ACM, 268–281.

[42] Fusang Zhang, Daqing Zhang, Jie Xiong, Hao Wang, Kai Niu, Beihong Jin, and Yuxiang Wang. 2018. From fresnel diffraction model to fine-grained human respiration sensing with commodity Wi-Fi devices. *Proc. ACM Interact. Mob. Wear. Ubiquitous Technol.* 2, 1, Article 53 (2018), 23 pages.

[43] Huan Zhang, Yiliang Jiang, Tao Jiang, and Hu Peng. 2021. Learn by referencing: Towards deep metric learning for singing assessment. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'21)*. 825–832.

[44] Xiaobin Zhuang, Huiran Yu, Weifeng Zhao, Tao Jiang, and Peng Hu. 2021. KaraTuner: Towards end to end natural pitch correction for singing voice in karaoke. *arXiv:2110.09121*. Retrieved from https://arxiv.org/abs/2110.09121

[45] Yongpan Zou, Haibo Lei, and Kaishun Wu. 2021. Beyond legitimacy, also with identity: Your smart earphones know who you are quietly. *IEEE Transactions on Mobile Computing* 22, 6 (2021), 3179–3192.

[46] Yongpan Zou, Jianhao Weng, Haibo Lei, Danyang Wang, Victor CM Leung, and Kaishun Wu. 2024. EarPrint: Earphone-based implicit user authentication with behavioural and physiological acoustics. *IEEE Internet of Things Journal* 11, 19 (2024), 31128–31143.

Received 2 May 2024; revised 28 September 2024; accepted 6 December 2024