MyMelody: An AI-Enhanced Unity Application for Vocal Range Identification, Real-Time Note Tracking and Performance Scoring in Accessible Singing Education

Lily Chen [1], Moddwyn Andaya [2]

[1] Westridge School, 324 Madeline Dr, Pasadena, CA 91105
[2] California State Polytechnic University, Pomona, CA 91768

ABSTRACT

MyMelody, a singing app created through Unity and C#, powered by an added element of AI, aims to become supplementary to traditional singing lessons, combating the limited accessibility, costliness, and other challenges people often face along their singing journey [2]. The three most important systems within the app are the Voice Type Identifier (for the user to identify their vocal range), Note Spawning (to display notes in the karaoke), and Song Scoring (for the user to determine their progress). The design process yielded many challenges, such as implementing math and outside algorithms to convert between pitch and frequency, making the UI fun and interactive for the user, and deciding how to score a user [1]. These challenges were addressed through iterative debugging and external research. System functionality was evaluated through a controlled test consisting of 10 predefined scenarios, of which the AI agent successfully handled 9. MyMelody is an innovative project that should be used increasingly, as it incorporates the power of artificial intelligence to help make music more accessible.

KEYWORDS

AI-Assisted Music Learning, Vocal Range Identification, Pitch Detection, Interactive Music Education

1. Introduction

Although music has become more accessible than ever, with every ad, TV show, mall, restaurant, public or private place blasting tracks from your newest favorite artist, music education remains largely neglected because of societal attitudes toward the pursuit of music. Many public-school organizations have reduced funding for music programs and teachers due to budget cuts and the deemed inessential nature of the fine arts, and "very little is being done at the university level to combat the [lack of music education] facing our schools" (source 1) [3]. As a result, most music education must be provided by families themselves [4]. This problem further increases the divide between different social classes and discourages people interested in music from pursuing their passion. Action must be taken to promote music accessibility to gain the benefits of music therapy and music's significant contributions to mental development; research has shown that learning music "facilitates learning other subjects and enhances skills that children inevitably use in other areas," (source 2) according to PBS News [5]. Even within the music industry, certain

David C. Wyld et al. (Eds): MLCL, CSITA, NMCO, AIRN, CONEDU – 2026
pp. 63-74, 2026. CS & IT - CSCP 2026 DOI: 10.5121/csit.2026.160105

instruments are less accessible than others; for example, voice lessons can be more costly and less accessible than piano lessons. Although public choirs provide a pathway to basic vocal training, learning in a large group with no individualized guidance may not be the most effective for certain singers who seek individual musical expression or personal development in music.

Aside from MyMelody, many other projects have tried to solve the lack of accessible vocal education. Their methodologies include using AI, providing online singing lessons, or traditional vocal training [6]. Although each approach has its own benefits, MyMelody has the greatest advantage overall.

Compared to the other AI approaches, MyMelody is more interactive and provides real-time feedback. It serves as a mentor, not a one-time guide. Users receive assistance from MyMelody's AI coach after each trial session instead of having to self-correct until they reach the optimal way of singing the song [7].

Although online singing lessons combine both technology and human teachings, MyMelody is still more accessible. As an AI, MyMelody can overcome both language barriers and scheduling conflicts while still providing real-time instruction. Furthermore, MyMelody does not face issues such as poor video or quality connection that may arise in other virtual settings.

At last, MyMelody shows its strength when compared to traditional vocal training in an in-person, one-on-one setting. Research has shown that traditional vocal training relies too much on "anecdotal" instruction, such as the teacher's own experiences, and not enough on evidence-based approaches. As an AI, MyMelody can analyze large amounts of data to find trends and the best methods for the user.

MyMelody is a virtual singing app that serves as a resource and guide to basic music education and vocal training. Users can view carefully constructed lesson plans to gain knowledge about healthy vocal technique, sing along to their favorite songs with backing tracks in keys adjusted to suit their vocal range and voice type, and receive feedback after their singing sessions to gauge where to improve. MyMelody uses an optimized OpenAI agent, modeled after a professional vocal coach, to analyze and score the user's performance on key factors such as pitch accuracy and note sustainment. Although the user will not be scored on dynamics or musicality, they will learn about these aspects through the lesson plans. MyMelody's use of AI sets it apart from other apps and solutions that offer free vocal training. The AI coach can collect data from the user over a span of time and detect patterns and inconsistencies, no matter how small (such as always falling flat on a certain note) that other coaches may overlook.

To evaluate the function of OpenAI in my app, I designed an experiment that input questions to the AI agent, Claire, which resided in the game's chatbox.

Question types ranged from asking about music theory to providing feedback on singing, to musically irrelevant topics; I expected Claire to respond appropriately to my input or redirect the conversation if it was off topic.

Overall, the OpenAI passed with a 90% success rate, demonstrating a solid usefulness in helping users with warmup exercises, music theory, and analysis of vocal improvement and consistent errors over time. Claire only failed when asked to generate a voice type for the user, as it was unable to access the voice-type identifier in the app and could not verify a voice type from the user's karaoke attempts. On the other hand, it can be argued that Claire should have passed this last case, as it was not programmed to access other elements of the app.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Saving & Scoring

MyMelody addresses concerns about handling multiple recordings of the same song by saving each recording as a separate entry rather than overwriting previous attempts. The AI analyzes all recorded performances to identify recurring patterns and common mistakes, allowing it to provide feedback informed by the user's full practice history. Although the system does not merge all past errors into a single cumulative summary, it leverages data from each attempt to guide users toward gradual and targeted improvement.

2.2. Pitch Detection and Gameplay

MyMelody's pitch detection may occasionally register background noise, as the system does not incorporate active noise cancellation. To address this, users are encouraged to monitor their performance through the karaoke interface, where filled-in note blocks visually indicate whether the correct pitch is being reached. In cases where sustained notes are not fully detected, the scoring system is intentionally forgiving, ensuring that minor detection issues do not significantly impact results. As a result, the overall score is designed to reasonably reflect a singer's true performance ability rather than being overly influenced by background noise.

2.3. UI

MyMelodyis designed with a clear and modular interface that allows users to navigate between the chat, karaoke, and lesson plans without the gamespace feeling crowded. The chat window is retractable and consistently positioned in the bottom-left corner, while the karaoke song list, lesson plans, and gameplay elements each occupy their own dedicated sections without overlapping. On smaller screens, the overall layout remains consistent, with only the dimensions and icon sizes scaling down to fit the display. Combined with the use of clear visual cues, colors, and sound feedback, the interface remains intuitive and user-friendly across different screen sizes.

3. SOLUTION

After first logging on to the game, the user is met with a pastel purple background and various UI elements they can choose from to advance the game. If the user is feeling ready to dive into the core of the game, they can click on the "Lessons" button or the "Song List" button in the middle of the page. Otherwise, the user can click on any of the three smaller icons at the bottom left that represent, from left to right, the chatbox (where users can converse with their AI coach, Claire), the settings (where users can change the volume or layout of the game), or the voice-type identifier (which can help users identify their vocal range). The smaller icons can be accessed from any point in the game, whereas the lessons and song list page contain information that is specific to their function.

The lessons page contains various short videos that instruct the user on basic music theory and healthy vocal technique. The videos were made using a human-designed curriculum but are presented through AI-generated voices and images using OpenAI's text-to-speech function [8]. These lessons serve as a guide to users who are more beginner singers and want to learn the basics before jumping into singing.

The song list page contains all the songs the user can choose to practice with. Marked on each individual song are buttons such as the semitone adjuster–which raises or lowers the key in which the user can sing to–the lightbulb UI, which gives the user a recommended list of vocal warmups they can do before singing the song–or the green play button, which, once clicked, advances the user to the karaoke page [9]. The user may start singing and notice that blocks will appear on the screen marked with note names; these indicate the notes of the song. If the user correctly hits the notes, the white arrow indicating their pitch will fall on the block, and the block will turn green. At the end of the song, the user is given a rating and feedback from AI. Every time the user sings, the game saves their progress.

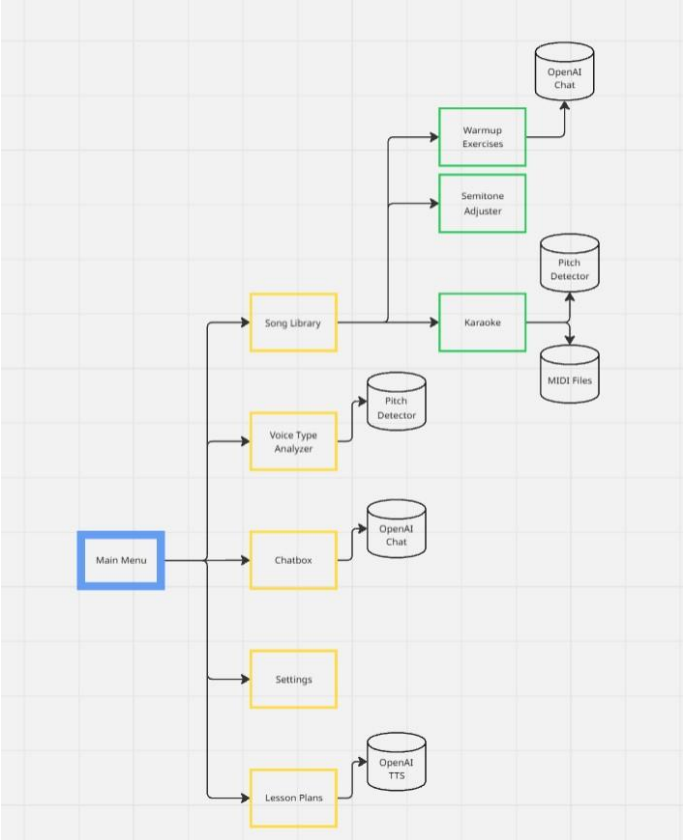

Figure 1. Overview of the solution

The voice type identifier script calculates and returns the user's voice type after collecting their highest note, lowest, note, and average speaking voice frequency. Calculating the user's voice type is essential to the rest of the game because users can adjust the key to fit their voice type in the karaoke.

This script relies on the voice type recorder script as well as a singleton class.

Figure 2. Find your vocal range

```
public int ReturnVoiceType(float lowFreq, float highFreq, float avgFreq)
{
    float bestDistance = float.MaxValue;
    int bestVoiceType = 2;
    foreach (var voicetype in voiceTypes)
    {
        if (lowFreq >= voicetype.lowFreq && highFreq <= voicetype.highFreq)
        {
            float midrange = (voicetype.lowFreq + voicetype.highFreq) / 2;
            if (midrange >= avgFreq && avgFreq >= 0.8 * voicetype.lowFreq)
                return voiceTypes.IndexOf(voicetype);
        }
        float distance = Mathf.Abs(lowFreq - voicetype.lowFreq)  + Mathf.Abs(highFreq - voicetype.highFreq);
        if (distance < bestDistance)
        {
            bestDistance = distance;
            bestVoiceType = voiceTypes.IndexOf(voicetype);
        }
    }
    return bestVoiceType;
}
```

Figure 3. Implementation of the voice type classification function

The ReturnVoiceType function takes in three values, lowFreq, highFreq, and avgFreq, which come from the user's recorded lowest and highest notes and average speaking voice. It begins by setting bestDistance to a very large number, which represents the current "closest match," and bestVoiceType to 2, the default voice type of Tenor [10]. The function then loops through each stored voice type, checking if the user's range falls completely between the range of a predetermined voice type using lowFreq and highFreq. If that condition is met, it calculates a midrange value, which is the average of the type's lowFreq and highFreq, and compares it to the voice type's avgFreq. If the midrange is greater than or equal to avgFreq, and the user's avgFreq is at least 80% of the type's lowFreq, then the function immediately returns that type as the best match. If the user's range does not fit cleanly within one type's range, the function instead calculates a distance, which is the sum of the differences between the user's lowFreq and the type's lowest frequency, plus the difference between the user's highFreq and the type's highest frequency. If this distance is smaller than the current bestDistance, the function updates bestDistance and sets bestVoiceType to the index of that type. After all types have been checked, the function returns the bestVoiceType, which will either be the closest match based on the smallest bestDistance.

Whenever the user start singing, the game show a "piano roll" that visually represents the song's notes as scrolling blocks. Each block corresponds to a specific pitch and duration, moving in

sync with the music. This helps players see timing and pitch accuracy while following the song in real time.

Figure 4. Karaoke pitch-tracking interface

```
foreach (var blockPlacement in blockPlacements)
{
    // Calculate Y position based on adjusted frequency
    float minFrequency = Note.OffsetFrequencyBySemitone(midiFileReader.minFrequency, blockSemitoneOffset);
    float maxFrequency = Note.OffsetFrequencyBySemitone(midiFileReader.maxFrequency, blockSemitoneOffset);

    float yPos = Note.MapFrequencyToPosition(blockPlacement.note.frequency, minFrequency, maxFrequency, pitchArrow.minArrowY, pitchArrow.maxArrowY);

    // Calculate the width based on the length (assuming length is in seconds and 100 units per second)
    float length = blockPlacement.endTime - blockPlacement.startTime;
    float blockWidth = length * 100f * widthStretchMultipliter;

    // Calculate the X position based on the starting time (assuming 100 units per second)
    float xPos = blockPlacement.startTime * 100f * widthStretchMultipliter;

    NoteBlock noteBlock = Instantiate(noteBlockPrefab, noteBlockParent);
    RectTransform noteTransform = noteBlock.GetComponent<RectTransform>();
    noteTransform.sizeDelta = new Vector2(blockWidth, noteTransform.sizeDelta.y);

    noteTransform.anchoredPosition = new Vector2(xPos + GetXOffset(), yPos);

    noteBlock.Init(blockPlacement);
    placedBlocks.Add(noteBlock);
}

1 reference
void UpdateBlockPositions()
{
    foreach (NoteBlock noteBlock in placedBlocks)
    {
        float startTime = noteBlock.blockPlacement.startTime;
        float timeDifference = startTime - currentTime;
        float xPos = timeDifference * 100f * widthStretchMultipliter + GetXOffset();

        RectTransform noteTransform = noteBlock.GetComponent<RectTransform>();
        noteTransform.anchoredPosition = new Vector2(xPos, noteTransform.anchoredPosition.y);
    }
}
```

Figure 5. Note block generation and positioning algorithm

This section covers the creation and movement of the note blocks to match the song. For each blockPlacement, the code generates a Y position by mapping the note's frequency between the lowest and highest frequencies for the song, adjusted for the semitone offset, into the pitch arrow's visible area. The width of the blocks will depend on the note's duration in seconds, multiplied by a scaling variable, and its position along the X axis is determined by when the note begins. A new NoteBlock prefab will be instantiated as a child underneath the parent UI, it will resize to match its note length, and any other special properties that might need to be set in reference to its X and Y position will be done at the time of block construction [14]. Every block will be created with the note block data and stored for future updates. During playback, UpdateBlockPositions will move the note blocks horizontally by determining the start time of each note block versus the song time, moving the note blocks left across the screen, and keeping to the rhythms of the song as it plays through.

The song scorer script listens to a song performance by the user, compare the user's pitch to the intended notes in the song, and records how well the user did. It then saves the performance stats and shows the score panel to the user.

This script relies on many other internal scripts to function, such as the pitch tracker, song roll player, saving system, score panel, and chat manager.

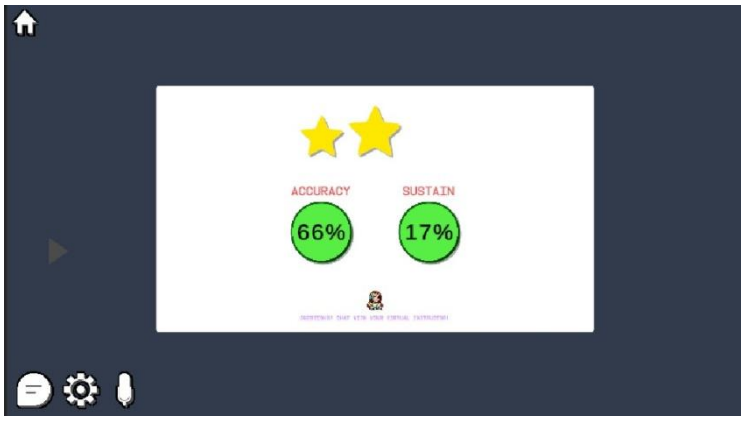

Figure 6. Performance feedback interface

```
float targetFrequency = currentBlock.blockPlacement.note.frequency;
float frequencyDiff = currentFrequency - targetFrequency;
float semitoneDiff = 12 * Mathf.Log(Mathf.Abs(frequencyDiff) < 0.0001f ? 0.0001f : Mathf.Abs(currentFrequency) / targetFrequency, 2);

currentData.detectedNote = Note.GetNoteFromFrequency(currentFrequency);
currentBlock.UpdateDetectedNote(currentData.detectedNote);
currentData.deviationMagnitude = semitoneDiff;

if (semitoneDiff > semitoneThreshold)
{
    currentData.frequencyDeviationHigh++;
}
else if (semitoneDiff < -semitoneThreshold)
{
    currentData.frequencyDeviationLow++;
}

bool isOnPitch = Mathf.Abs(semitoneDiff) <= semitoneThreshold;

if (isOnPitch)
{
    currentData.isOnTime = true; // Mark once if we ever hit the note correctly
    currentData.sustainDuration += Time.deltaTime;
    currentBlock.UpdateSustainBar(currentData.sustainDuration);
}

var lastBlock = songRoll.placedBlocks[^1];
float lastNoteEndTime = lastBlock.blockPlacement.endTime;

if (songRoll.currentTime >= lastNoteEndTime + 1)
{
    StopScoring();
}
```

Figure 7. Pitch deviation scoring logic

In the karaoke portion of the game, the user's goal is to match the pitch of the moving blocks on the screen. Each block has a targetFrequency, and the system compares this with the user's currentFrequency. The difference between them is first measured directly (frequencyDiff) and then converted into a more musically useful value, semitoneDiff, using a logarithmic formula. This expresses the pitch gap in terms of semitones rather than raw frequency. The user's pitch is also translated into a musical note (e.g., "A#5" or "D4") through GetNoteFromFrequency, and that detected note is displayed on screen with UpdateDetectedNote. The semitone difference is stored in deviationMagnitude and serves as the basis for scoring.

The system uses a semitoneThreshold of 2, meaning the user can be up to two semitones sharp or flat and still be counted as "on pitch." If the user drifts more than two semitones above,

frequencyDeviationHigh is incremented; if more than two semitones below, frequencyDeviationLow is incremented. When the user's pitch is within the threshold, they are marked as on-time, their sustainDuration increases frame by frame, and the sustain bar on the current note block on the user's screen turns green to reflect their accuracy. Finally, once the last note in the song has finished and one extra second has passed, the system ends scoring.

4. EXPERIMENT

I will be testing out the function of the OpenAI agent in MyMelody'schatbox. The agent, which serves as the user's personal vocal coach, should adhere to the topics of music theory regardless of the user's input. It should also be able to provide relevant feedback upon analyzing the user's singing data. The significance of this experiment lies in providing a wide range of users with the most appropriate and helpful responses.

In this experiment, I will ask a series of 10 questions to the chatbox, with topics ranging from requesting assistance in specific areas to music theory to completely unrelated subjects. I will also visit the song library and try singing along with the virtual karaoke; afterward, I will ask the AI agent to provide feedback and determine the quality of its feedback based on my performance (the number of notes I hit and how long I sustained them for). Ultimately, I will compare the AI's answers to my expectations and attempt to identify patterns where they diverge. I believe the AI can be considered functional if it adheres to expectations at least 80% of the time, and if there is no obvious correlation between the answers that do not pass.

Table 1: This table summarizes the ten evaluation scenarios, including user prompts, expected behavior, observed system responses, and pass/fail outcomes. Karaoke-based scenarios were evaluated based on whether the AI agent referenced pitch accuracy, note sustain-duration, or recurring patterns consistent with the performance metrics generated by the scoring system.

Test	Prompt	Expectation	Result	Pass
1	Can you give me some exercises to improve my singing volume?	Humming, Sirens, Lip trills, etc. + an explanation of why it improves volume	Diaphragmatic Breathing + explanation, Lip Trills + explanation, Sustained Breathing + explanation	Yes
2	What are the nearest directions to Disneyland?	It will say it cannot help with that	It is not sure what I am asking and tries to redirect the conversation to vocal training	Yes
3	I'm having trouble reaching high notes. What can I do?	It will provide exercises or a general approach	Siren exercise + advice to relax and not strain	Yes
4	What is a mixed voice? Follow-up: How do I develop it?	It will explain that mixed voice merges both head and chest voice; it will provide exercises to strengthen mixed voice	Explains the definition and usefulness of mixed voice; recommends the nay-nay-nay exercise	Yes
5	Karaoke Test #1 with Here With Me	It will show pitch accuracy, sustain, and other feedback	Shows pitch accuracy, Sustain, and Overall Performance. Specified particular issues with sustaining notes and higher ranges and comfortability with lower ranges	Yes
6	Karaoke Test #2 with Here With Me Follow-up: What improved from Session 1?	It will provide feedback in the same areas as Test #1; it will compare Sessions 1 and 2	Shows problems with Sustain; Describes improvement in certain high notes from Session 1 (ex: A#4)	Yes

7	Karaoke Test #3 with Here with Me Follow-up: What are some general patterns you are noticing in my performances?	It will provide feedback in the same areas as Tests #1 and #2; it will make generalizations from all three sessions	Notices an improvement in Sustain but problems with G#4 and D#4; recommends Siren exercises for high notes; notices general challenges with sustain, high notes, and certain notes; notices mid-range stability, pitch improvement, and acknowledges me actively seeking areas to improve	Yes
8	How do you assign a grade? What is the criteria?	Pitch, sustain, etc.	Pitch, sustain, rhythm + timing, overall vocal control; emphasizes that it provides feedback and not a quantitative "grade"	Yes
9	Who is Mariah Carey?	It will tell me who she is and her contributions to music	Biography + her accomplishments in singing + what makes her a great singer	Yes
10	What is my voice type?	It should be able to calculate it based on previous data and singing sessions	It provides insight but cannot narrow down to a specific voice type	No

Table 2: This table provides the results of Table 1 but groups them more cleanly into query categories.

Category	Number of Tests	Pass Rate
Music Theory & Vocal Technique Queries	4	100%
Karaoke Performance Feedback	3	100%
Off-topic Prompt Redirection	2	100%
Cross-module Inference (voice-type analyzer)	1	0%
Overall	10	90%

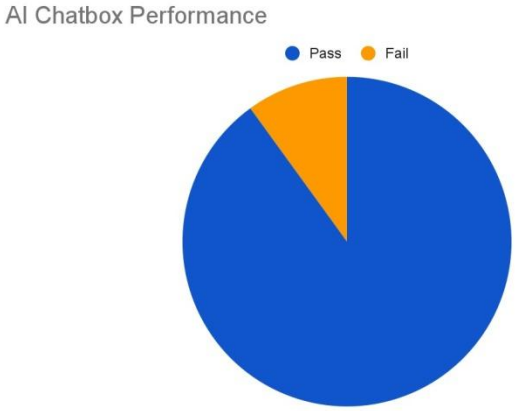

Figure 8. AI chatbox performance

MyMelody'sOpenAI agent successfully executed 9 out of 10 inputs, giving it an accuracy rating of 90%. It is therefore functional. The agent was able to answer any questions about music theory and vocal warmups. It could also advise before the user starts singing and provide

feedback after singing. Most importantly, the agent could analyze trends and preserve the user's previous attempts at the karaoke to track progress over time. Nevertheless, it failed when I asked for my voice type. This task required direct access to the voice-type identification module of the game, which is isolated from the AI system. As a result, the agent could not get a definitive voice classification from karaoke performance data alone. This limitation reflects a deliberate system design choice prioritizing data privacy rather than a failure of instructional capability.

5. RELATED WORK

In 2021, the Institute of Electrical and Electronics Engineers published a research paper titled "Application of AI as a Singing Trainer," which outlines an approach through Artificial Intelligence to provide users with proper vocal training [11]. The Institute's AI takes a song selected by the user, alongside voice clips from the user, to create a new version of the song sung in the user's voice. This mashup serves as the optimal way for the user to sing the song.

The Institute claims the user can compare their current progress to the mashup to find and correct differences, helping them improve their mistakes. Although this approach seems promising, it offers no real guidance. The user must self-correct their mistakes, and the AI provides no written feedback. Furthermore, the quality of the recording itself cannot be ensured; it may sound robotic and not take into account the multitude of ways the user can sing the song.

In contrast, MyMelody's AI can analyze the user's voice after each recording, providing realtime feedback. It keeps track of their progress over time and consistently coaches, while the Institute's AI merely provides a one-time guide. The colorful UI of MyMelody and conversational style AI further enhance MyMelody's advantage. It should be noted that the AI component in MyMelody is not responsible for rule-based pitch detection; instead, it functions as an instructional feedback agent that interprets performance summaries.

In their paper titled "Exploring teachers' and pupils' behaviour in online and face-to-face instrumental lessons," authors Andrew King, Helen Prior, and Caroline Waddington-Jones discuss their collaboration with North Yorkshire Music Action Zone and YouCan Play to bring online singing lessons to students in remote, rural England [12].

They provided voice teachers with advanced technologies, such as the Roland AV Mixer (a camera with 3 angles and good quality sound), and conducted lessons over Skype. Results from analyzing the lessons and surveying members of the trial found challenges with video and connection quality due to the online environment, as well as teachers reporting increased teaching challenges. Nevertheless, students focused well and made good progress.

Although this approach is quite strong and blends technology with traditional teaching, MyMelody has an advantage due to its strength as an AI. MyMelody is available anytime, anywhere, and students don't need to rely on specific teacher availability. MyMelody also helps overcome language and cost barriers, especially to underserved communities. Finally, MyMelody can avoid some of the challenges faced in online lessons, such as a bad connection or video quality.

Laura Crocco's thesis, titled "A Systematic Approach to One-to-One Classical Singing Training in Higher Education," aims to address an error in traditional vocal training [13].

Crocco claims that classical vocal genres take years to learn and are highly demanding on a singer's physical and emotional well-being; yet, the way classical singers are taught does not rely on scientific or systematic methods but instead on the instructor's personal experience, making

the process of learning all the more difficult. The study found that while instructors commonly rely on anecdotal instruction, feedback, and modelling, they tend to underuse motivation, explanation, and perceptual training as well as evidence-based approaches.

MyMelody solves this problem because of its pattern-tracking abilities and motivational style. As an AI, MyMelody can contribute to storing large amounts of data and finding the best methods through data analysis. MyMelody's programming as a friendly, amiable instructor can also make up for the lack of motivation and explanation in traditional training.

6. CONCLUSIONS

Although MyMelody has made vocal training free and more accessible, it is currently limited to computer and PC use only. This may feel inconvenient for those who want to practice with their phone or on the go. Another limitation of MyMelody comes from the fact that the app is still in its early stages. Users are currently limited in the number of songs they can sing, as the songs need to be imported manually. MyMelody cannot connect with apps such as Spotify or YoutubeMusic to access a larger song bank. This limitation is currently being worked on.

Additionally, while the AI lesson plans offer guidance, an area for improvement is to replace the AI instructors with human instructors and implement their curricula [15]. This could enhance engagement and provide a more interactive and personalized learning experience for the user.

REFERENCES

[1] Orhani, Senad. "Incorporating algorithms into mathematics syllabus." International Journal of Curriculum Development, Teaching and Learning Innovation 3.2 (2025): 88-100.
[2] Han, Yang. "Using mobile applications in the study of vocal skills." Education and Information Technologies 28.2 (2023): 2107-2127.
[3] Abril, Carlos R., and Brent M. Gault. "The state of music in secondary schools: The principal's perspective." Journal of Research in Music Education 56.1 (2008): 68-81.
[4] Green, Lucy. "Popular music education in and for itself, and for 'other'music: Current research in the classroom." International journal of music education 24.2 (2006): 101-118.
[5] Hillecke, Thomas, Anne Nickel, and Hans Volker Bolay. "Scientific perspectives on music therapy." Annals of the New York Academy of Sciences 1060.1 (2005): 271-282.
[6] Mendes, Ana P., et al. "Effects of vocal training on the acoustic parameters of the singing voice." Journal of voice 17.4 (2003): 529-543.
[7] Holland, Simon. "Artificial intelligence in music education: A critical review." Readings in music and artificial intelligence (2013): 239-274.
[8] Suni, Alfa Faridh, et al. "Text-to-Speech User Interface for ChatGPT." 2024 4th International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS). IEEE, 2024.
[9] Felekoglu, Elvan, et al. "Effect of karaoke training applied with telerehabilitation-based exercise training on health-related outcomes in children with asthma." European Journal of Pediatrics 184.9 (2025): 1-13.
[10] Titze, Ingo R., Sharyn Mapes, and Brad Story. "Acoustics of the tenor high voice." The Journal of the Acoustical Society of America 95.2 (1994): 1133-1142.
[11] Vinze, Vivek, et al. "Application of AI as Singing Trainer." 2021 International Conference on Advances in Computing, Communication, and Control (ICAC3). IEEE, 2021.
[12] King, Andrew, Helen Prior, and Caroline Waddington-Jones. "Exploring teachers' and pupils' behaviour in online and face-to-face instrumental lessons." Music education research 21.2 (2019): 197-209.
[13] Crocco, Laura. A Systematic Approach to One-to-One Classical Singing Training in Higher Education. Diss. 2018.

[14] Liu, Yike, et al. "NoteBlock: Prototype Design of Music Learning Experience for Blind and Low Vision Children in Preschool Ages." Companion of the 2024 on ACM International Joint Conference on Pervasive and Ubiquitous Computing. 2024.

[15] Van den Berg, Geesje, and Elize Du Plessis. "ChatGPT and generative AI: Possibilities for its contribution to lesson planning, critical thinking and openness in teacher education." Education Sciences 13.10 (2023): 998.