A Piano Practice Evaluation Tool Based on the Onset Detection, Pitch-Detection and Weighted DTW Algorithm

Yujie Liu
Master of Machine Learning
yujieliu@kth.se

Yizhen Zhao
Master of Wireless Communication
yizhen@kth.se

Jiarui Su
Master of Wireless Communication
jiaruis@kth.se

Yilin Qiu
Master of Information and Network Engineering
yilinqiu@kth.se

ABSTRACT

Providing timely feedback during piano practice is crucial for beginners to enhance their performance. This project developed an application that compares user-submitted practice recordings with professional audio samples to evaluate performance based on similarity.

Due to the complexity of piano audio, various methods for note extraction were explored. Ultimately, we combined onset detection with the Constant-Q Chromagram to effectively capture the main melody. To assess similarity, we utilized dynamic time warping, and further improved accuracy by incorporating weighted adjustments into the calculations.

Code: https://github.com/vicky-liu17/Music-Informatics

1. INTRODUCTION

1.1 Background

It has become a common phenomenon for parents to send young children to study music. They believe that learning music, especially instruments such as piano, violin, and guitar, can develop children's intelligence and improve their overall abilities [1]. According to research by the Associated Board of the Royal Schools of Music [2], approximately 69% of children in the United Kingdom reported that they are currently learning a musical instrument, with more than half of them attending courses at music training institutions. In China, although only 2.5% of the population is involved in arts education, far lower than the 20% in developed countries, the market size is still enormous due to the large population base. Among the participants, 75% are teenagers aged 5 to 15 years old. In 2014, the market size reached 11.4 billion yuan. As parents are placing increasing importance on music education, this number continues to grow [3].

It is widely recognized that learning musical instruments requires extensive practice. In previous surveys, experienced violinists reported having practiced for an average of more than 5,000 hours, with some exceeding 10,000 hours. It is commonly believed that to become an expert, one needs to spend at least 10,000 hours practicing [4]. This is especially important for young children, as practice plays a vital role in both their music learning and brain development [5]. In addition to absorbing knowledge in music classes, they also need timely supervision and guidance during daily practice [6].

However, without proper guidance and timely feedback, beginners often struggle to assess their own progress, which can lead to reinforcing incorrect techniques or misunderstandings. Over time, these issues may hinder development and affect motivation.

1.2 Goal

This project aims to analyze the similarity between beginner performance recordings and professional samples. The application is designed to be simple and user-friendly, while ensuring the results are as accurate as possible. A frontend interface will be built using Python's Tkinter, providing users with an intuitive platform. The algorithm will extract the main melody of the piano piece, compare it with the sample, and display the results on the interface for easy evaluation.

1.3 Melody Extraction

Piano compositions are inherently complex, often featuring elements such as chords and intricate harmonic progressions. This project, however, focuses on simplifying this complexity for beginner-level analysis by extracting and analyzing only the main melody.

Melody extraction is a significant research topic in the field of Music Information Retrieval (MIR). Its goal is to extract a sequence of fundamental pitches from a musical fragment, which listeners perceive as the "essence" of the music [7].

Melody extraction requires two main steps: first, segmenting the notes from the musical piece, and then extracting the pitch of each note.

For note segmentation, commonly used algorithms include onset and offset detection methods, which are widely applied in musical signal processing. In this project, we evaluated several note segmentation approaches, including a simple threshold-based onset and offset detection algorithm, spectral flux detection, and machine learning techniques. Ultimately, we selected the spectral flux detection

algorithm, as it provided an optimal balance between accuracy and computational efficiency.

For pitch extraction, various methods can be used, including the Fast Fourier Transform, Constant-Q Transform, and machine learning-based approaches. In this project, we chose the Constant-Q Transform (CQT) because of its effectiveness in analyzing musical signals with varying frequency resolutions, making it particularly well-suited for pitch extraction in complex audio signals. Additionally, we opted for CQT due to the limited computational power available for running some machine learning models [8].

1.4 Melody Analysis

For melody analysis, we apply Dynamic Time Warping (DTW) to compare the extracted melody from the performance with a reference sample. DTW is a widely used algorithm for measuring similarity between two sequences that may differ in timing or speed, making it ideal for evaluating musical performances where tempo variations are common.

In this project, we extended the traditional DTW algorithm by incorporating loudness-based weighting. Rather than treating all points in the sequence equally, we assign weights based on the loudness of each note. This modification allows louder, more prominent notes to have greater influence in the comparison process. Given the distinctive dynamic contrasts in piano music, where variations in loudness are critical for conveying musical expression, this weighted approach results in a more musically informed alignment. Consequently, the enhanced DTW provides a more accurate comparison of the performed melody against the reference, improving the overall assessment of both musical accuracy and expressive quality[9].

2. METHOD

2.1 Piano Note Detection Using FFT with Threshold-Based Onset Segmentation

In this method, we employ an FFT-based approach combined with threshold-based segmentation for detecting piano notes and extracting their corresponding pitches. The process consists of two primary steps: signal segmentation using amplitude thresholds for note detection and frequency-domain analysis using FFT to extract the pitch of each detected note.

2.1.1 Signal Segmentation and Note Onset Detection

To identify individual notes within the piano audio signal, we implement a threshold-based onset detection technique. This method divides the audio into segments by detecting points where the amplitude of the signal rises above a defined threshold (onset) and falls below another threshold (offset). These thresholds are calculated as a proportion of the maximum amplitude of the audio signal:

$$thresh_{up} = 0.2 \times \max(audio\_signal) \tag{1}$$

$$thresh_{down} = 0.04 \times \max(audio\_signal) \tag{2}$$

This allows for isolating each note's boundaries, making the following frequency analysis more efficient.

2.1.2 Frequency Analysis Using FFT

For each identified note segment, we apply the Fast Fourier Transform (FFT) to convert the time-domain signal into its frequency-domain representation. The FFT returns a spectrum of frequencies present in the note, from which we can determine the dominant frequency by identifying the peak in the magnitude spectrum. The dominant frequency f is calculated as:

$$f = \frac{I \times F_s}{N} \tag{3}$$

Where:

- I is the index of the frequency bin with the maximum magnitude,
- F_s is the sampling rate of the audio signal,
- N is the number of points in the FFT.

This dominant frequency is assumed to correspond to the fundamental frequency (pitch) of the note.

2.1.3 Pitch to MIDI Conversion

To standardize the detected pitch, we convert the frequency into a MIDI note number, which is a widely-used digital representation of musical notes. The MIDI note number is computed as:

$$MIDI = pitch + 12 \times (octave + 4) \tag{4}$$

Where:

- *pitch* represents the musical note class (C, C, D, etc.),
- *octave* refers to the frequency range in which the note resides.

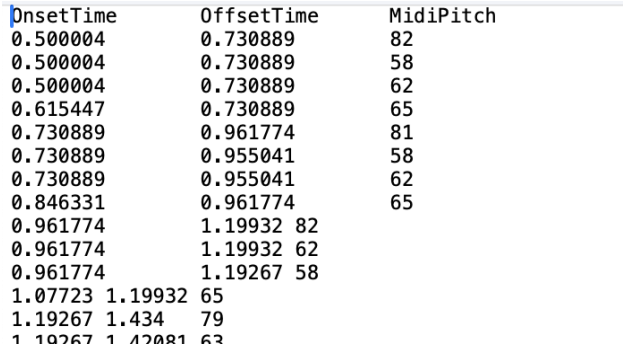
This formula adjusts the pitch class and octave to fit within the MIDI scale, providing a standardized output that can be used for further musical analysis or transcription.

2.2 Librosa-Based Onset and Pitch Detection

This method utilizes the librosa library for detecting note onsets and extracting pitches from the audio signal using the Constant-Q Transform (CQT).

In the first step, onset detection is performed using librosa method, which identifies peaks in an onset strength envelope. This envelope captures sudden increases in energy or changes in spectral content across frames, which often signal the start of a new note. By using a peak-picking algorithm, librosa isolates frames with significant energy increases, effectively marking potential note onsets. Configurable parameters like minimum separation between peaks and amplitude thresholds help refine the accuracy of onset detection, ensuring that only the most prominent onsets are selected.

After identifying onset times, the Constant-Q Transform (CQT) is applied to detect the dominant pitch at each onset.



OnsetTime	OffsetTime	MidiPitch
0.500004	0.730889	82
0.500004	0.730889	58
0.500004	0.730889	62
0.615447	0.730889	65
0.730889	0.961774	81
0.730889	0.955041	58
0.730889	0.955041	62
0.846331	0.961774	65
0.961774	1.19932 82	
0.961774	1.19932 62	
0.961774	1.19267 58	
1.07723 1.19932 65		
1.19267 1.434 79		
1.19267 1.42081 63		

Figure 1. TXT label files in the MAPS dataset

Unlike the Fast Fourier Transform (FFT), which provides a linear frequency scale, CQT uses a logarithmic frequency scale that aligns more closely with musical intervals, where each octave doubles in frequency. This logarithmic resolution is particularly advantageous in music analysis, as it matches how the human ear perceives pitch: higher resolution for lower frequencies and lower resolution for higher frequencies. This alignment allows CQT to represent musical notes more naturally, making it ideal for detecting pitches across different octaves.

2.3 Machine Learning Method

In this study, we developed a machine learning approach for pitch recognition and transcription, using the MAPS dataset as the primary training resource. The MAPS dataset is specifically tailored for automatic music transcription tasks, offering a range of unique advantages. It features rich polyphonic audio recordings and includes accompanying MIDI and text (TXT) files as labels, which makes it well-suited for supervised learning applications. The MIDI and TXT labels provide precise information on note onsets, offsets, and pitches, offering a reliable ground truth for training and evaluating models in tasks like pitch detection and multipitch transcription.

We experimented with both a baseline model and a DNN, though the DNN has yet to complete training due to time and computational constraints.

2.3.1 Baseline Algorithm

In this method, in addition to using the constant-Q features extracted from the audio data, which provide a logarithmic frequency representation, we also applied the One-vs-Rest (OvR) approach, a common strategy suitable for multiclass classification problems.

2.3.2 One-vs-Rest Approach

The *One-vs-Rest (OvR)* approach is a widely used strategy for solving multiclass classification problems using binary classifiers. In a multiclass problem with K classes, the OvR approach trains K binary classifiers. Each classifier is trained to distinguish one class from all the others, transforming the multiclass problem into multiple binary classification problems.

For each class k (where $k \in \{1, 2, \ldots, K\}$), a binary classifier is trained to differentiate class k (positive class) from all other classes $\neq k$ (negative class). This can be formalized as follows:

Let $X \in R^{n \times m}$ be the feature matrix with n samples and m features, and let $y \in \{1, 2, \ldots, K\}$ be the label set with K possible classes. For each class k, the logistic regression classifier is defined by the following probability model:

$$P(y = k \mid x) = \frac{1}{1 + \exp\left(-(\beta_k^T x + \beta_{0k})\right)} \tag{5}$$

where:

- β_k is the vector of weights (parameters) for class k,
- β_{0k} is the bias term (intercept) for class k,
- $x \in R^m$ is the feature vector of the input sample.

The classifier is trained to maximize the likelihood that the model correctly classifies samples of class k versus all other classes.

Once all K binary classifiers are trained, each classifier outputs a probability score for its respective class. During the prediction phase, the model assigns a sample to the class with the highest predicted probability. The final predicted class $\hat{y}$ is given by:

$$\hat{y} = \arg \max_{k \in \{1,\ldots,K\}} P(y = k \mid x) \tag{6}$$

Thus, the class with the highest score is selected as the predicted label for the given sample.

In summary, the OvR approach effectively decomposes a multiclass classification problem into multiple binary problems. Each classifier predicts whether a sample belongs to a specific class or not, and the final decision is made by selecting the class with the highest probability score.

2.3.3 Deep Neural Network

The Deep Neural Network method in this approach is based on constructing a deep neural network (DNN) using a varying number of layers to evaluate performance across different model architectures. The network employs a cyclical learning rate (CLR), which oscillates between a minimum and maximum value to improve convergence and avoid local minima during training. Each model is trained with binary cross-entropy loss, as the task involves multi-label classification (detecting multiple musical notes simultaneously). After training, the model's performance is evaluated using metrics such as precision, recall, accuracy, and the F1-score.

2.4 Dynamic Time Warping (DTW) Algorithm

Dynamic Time Warping (DTW) is a widely used algorithm for measuring the similarity between two time-dependent sequences that may differ in speed or timing. It is particularly useful in tasks such as speech recognition, music transcription, and time-series analysis, where temporal misalignment exists between two sequences.

Given two sequences:

$$A = [a_1, a_2, \ldots, a_n] \quad and \quad B = [b_1, b_2, \ldots, b_m]$$

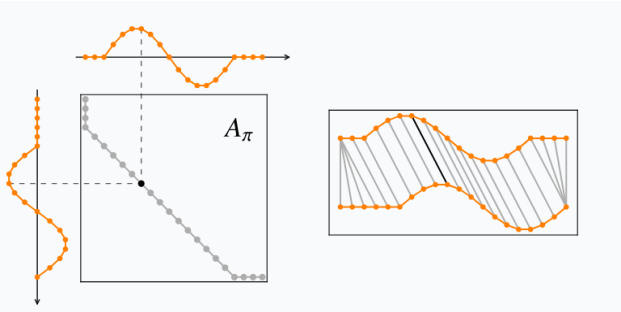


Figure 2. DTW Algorithm

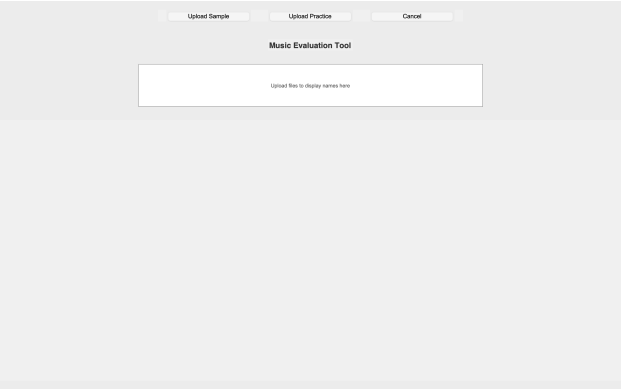


Figure 3. Interface

where A and B are of lengths n and m, respectively, DTW seeks to find the optimal alignment that minimizes the cumulative distance between the sequences.

The algorithm starts by computing a *distance matrix* D, where each element $D(i, j)$ represents the distance between points a_i and b_j. Typically, the Euclidean distance is used:

$$d(a_i, b_j) = (a_i - b_j)^2 \tag{7}$$

Once the distance matrix is computed, the algorithm constructs an *accumulated cost matrix* C, where each element $C(i, j)$ represents the minimum cumulative cost to align the subsequences A_1^i and B_1^j. The recurrence relation for calculating C is given by:

$$C(i,j) = d(a_i, b_j) + \min\{C\,(i-1,j)C(i,j-1)C(i-1,j-1) \tag{8}$$

The boundary conditions for initializing the matrix are:

$$C(1,1) = d(a_1, b_1) \tag{9}$$

$$C(i,1) = \sum_{k=1}^{i} d(a_k, b_1), \quad C(1,j) = \sum_{k=1}^{j} d(a_1, b_k) \tag{10}$$

Finally, the DTW distance between the sequences is given by the value at the last position of the accumulated cost matrix $C(n, m)$, representing the minimum cumulative distance for aligning the two sequences:

$$DTW(A, B) = C(n, m) \tag{11}$$

The optimal alignment path can be retrieved by tracing back from $C(n, m)$ to $C(1, 1)$ following the minimum cost path.

2.4.1 *Weighted DTW Algorithm*

In piano performance, different notes vary in strength and prominence, with stronger notes often playing a more important role. This section introduces a weighted one-dimensional Dynamic Time Warping (DTW) algorithm, designed to align onset times, strength, and chroma values between a sample and a practice sequence. By incorporating onset strength as a weight in the distance calculation, this approach allows more prominent notes to have a greater influence in the alignment process.

- **Data Preprocessing**: First, the onset times, strengths, and chroma values for both sample and practice sequences are transformed into one-dimensional sequences. This transformation produces feature sequences for both sample and practice data, as well as corresponding strength sequences, which are prepared for subsequent DTW calculations.

- **Strength Normalization**: To ensure consistent weighting during distance calculation, strength values in both sequences are normalized to lie within a standardized range of 0 to 1. This normalization enables a balanced contribution of strength values across the two sequences.

2.4.2 *Weighted Distance Function*

The central feature of this method is the weighted Euclidean distance function, which adjusts the distance between two points based on onset strength. For chroma values x and y, with corresponding strengths s_x and s_y, the weighted distance function is defined as:

$$d(x, y, s_x, s_y) = \|x - y\| \cdot \left(1 + \alpha \cdot \frac{s_x + s_y}{2}\right) \tag{12}$$

where α is a user-defined weight parameter. By assigning a larger weight to points with higher onset strength, this distance function enhances the influence of prominent onsets on the overall DTW alignment.

2.4.3 *DTW Distance Calculation*

The DTW distance is calculated by summing the weighted distances along the optimal alignment path between the sample and practice sequences. For each pair of matched points, the algorithm applies the weighted distance function, which takes into account both chroma similarity and onset strength. This approach results in a total DTW distance that gives perceptually significant onsets—those with stronger intensity—greater influence on the alignment outcome, enhancing the accuracy of onset matching.

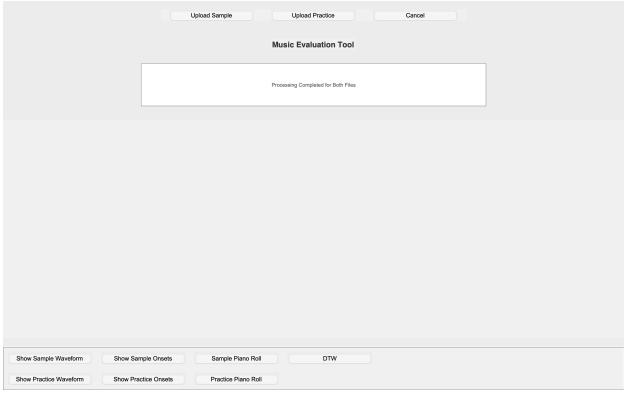


Figure 4. Interface

3. RESULTS

3.1 User Interface

The app begins with a simple page that includes three main buttons(Figure 3):

Upload Sample: allows users to upload a reference audio file that serves as the comparison standard.

Upload Practice: enables users to upload their practice audio, which will be compared to the reference file for evaluation.

Cancel: provides an option to cancel the current operation or reset the uploaded files.

These buttons serve as the primary controls for initiating the music evaluation process.

Once both files have been successfully uploaded, additional buttons appear at the bottom of the interface. These new buttons allow the user to visualize various aspects of the uploaded audio files.(Figure 4)

3.2 Melody Extraction

3.2.1 Piano Note Detection Using FFT with Threshold-Based Onset Segmentation

We first take "Shufflin' Along" by John Thompson, a commonly used practice piece for beginner piano learners, as an example. The sheet music is simple, with clear notes, to test the effectiveness of this algorithm.

In this paper, we use evaluation metrics such as Precision P, Recall R, and F-Measure F to evaluate the performance of note onset detection. The F-Measure is a comprehensive evaluation metric that combines Precision and Recall. The formulas for Precision P, Recall R, and F-Measure F are given as follows:

$$R = \frac{N_{correct}}{N_{correct} + N_{miss}} \times 100\% \tag{13}$$

$$P = \frac{N_{correct}}{N_{correct} + N_{error}} \times 100\% \tag{14}$$

$$F = \frac{2 \times P \times R}{P + R} \times 100\% \tag{15}$$

In above equations, $N_{correct}$ represents the number of correctly detected notes, N_{miss} represents the number of

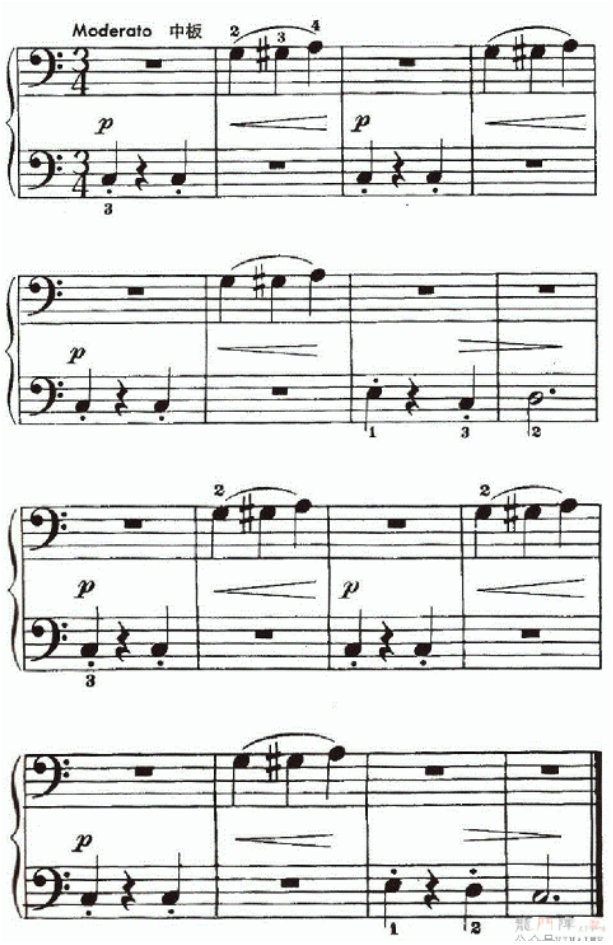


Figure 5. The sheet music for "Shufflin' Along"

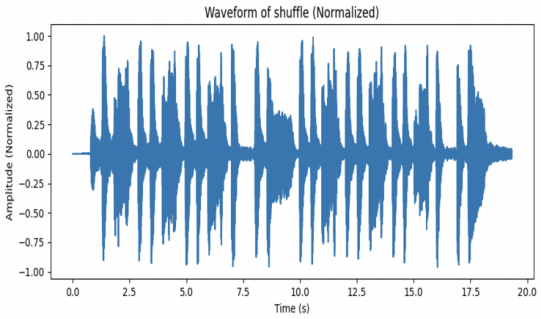


Figure 6. Waveform of "Shufflin' Along" sample

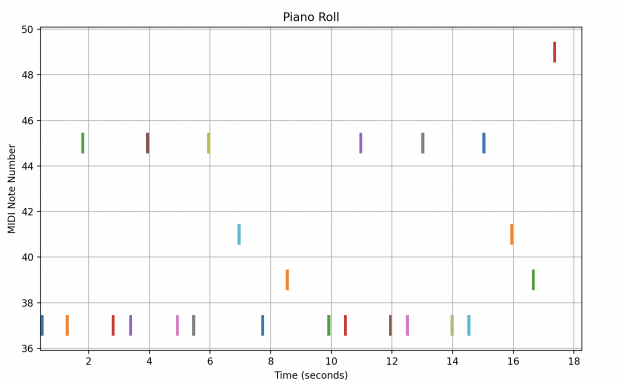


Figure 7. Piano Roll of "Shufflin' Along" sample

notes that were missed during detection, and N_{error} represents the number of incorrect notes detected.

The piece contains a total of 36 notes, but this algorithm was only able to identify 24 of them. The evaluation results show a Recall (R) of 62.5%, a Precision (P) of 83.33%, and an F-Measure (F) of 71.43%, which are not ideal. Subsequently, we conducted experiments using "Two Tigers" and "Canon," with F-Measure (F) results of 68.32% and 64.03%, respectively.

During this process, many notes were missed, especially when we tested with "Flight of the Bumblebee," where all the notes were connected together, resulting in only one note being detected in the end.

After analysis, the poor performance of this algorithm may be due to the following reasons:

Threshold-Based Detection: The `noteparse` function uses fixed thresholds (`threshup` and `threshdown`) to detect the start and end of notes. In pieces with rapid transitions and minimal amplitude variation, these thresholds may fail to capture each note accurately, causing the algorithm to group multiple notes into a single one.

Limitations of FFT and Bandpass Filtering: Each note window is processed through a Fast Fourier Transform (FFT) and bandpass filtering to detect its dominant frequency. However, in fast-paced passages, this approach can blur note boundaries, especially when multiple frequencies overlap within a short time window. The FFT analysis may not distinctly resolve each note, leading to inaccurate frequency and onset detection.

Fixed Frequency Mapping: The `findpitch` function determines pitch based on octave and frequency ranges. However, in fast sequences, overlapping harmonics may interfere with pitch detection, resulting in inaccurate note assignment. In "Flight of the Bumblebee," where notes transition rapidly and frequencies overlap, the algorithm may incorrectly merge several pitches into a single note.

3.2.2 Baseline logistic regression machine learning algorithm

We divided the downloaded MAPS dataset into three parts: train, test, and validation, and performed testing after training was completed. The classification report provides the following averages: macro average with a precision of 0.20, recall of 0.69, and F1-score of 0.28; weighted average with a precision of 0.38, recall of 0.90, and F1-score of 0.52; and samples average with a precision of 0.25, recall of 0.85, and F1-score of 0.37.

The primary reasons for the low accuracy observed are as follows:

- **Insufficient Training Data**: With only around 20 training pieces in a single MAPS folder, the sample size may be inadequate for effective model training. Limited data can lead to issues such as overfitting, where the model learns patterns specific to the training data but struggles to generalize to new samples.

- **Class Overlap**: In music, audio features for different notes frequently overlap due to harmonics and similar spectral content. `OneVsRestClassifier` uses a separate binary classifier for each class, but with overlapping classes, distinguishing between similar notes becomes challenging, which increases the likelihood of misclassification.

- **Complexity and Scalability**: For a large number of notes (classes), `OneVsRestClassifier` requires training a separate model for each note. This process is computationally expensive and time-consuming, especially when using high-dimensional audio features like CQT, making it inefficient as the number of classes or feature complexity grows.

3.2.3 Librosa-Based Onset and Pitch Detection

This is the solution we ultimately chose to adopt. Using *Shufflin' Along* as an example, the algorithm detected 40 dominant pitches. However, upon reviewing the generated piano roll, we found that it misinterpreted some of the longer-duration notes at the beginning and end of the piece as multiple notes. The F-Measure (F) reached 92.43%. When tested with *Two Tigers* and *Canon*, the F-Measure (F) values were 91.93% and 87.02%, respectively.

The high accuracy of `librosa`'s onset and pitch detection can be attributed to several advanced techniques:

1. **Advanced Onset Detection Algorithms**: `librosa` utilizes sophisticated onset detection methods that analyze amplitude, spectral flux, and energy changes within the audio signal. By detecting sudden changes in these characteristics, the algorithm can accurately pinpoint note onsets, even in fast and complex passages.

2. **Constant-Q Transform (CQT) and Spectral Analysis**: `librosa` employs the Constant-Q Transform (CQT) to capture detailed frequency information over time. The CQT's frequency resolution, particularly suited to musical scales, enhances pitch detection accuracy by offering finer resolution for lower pitches.

3. **YIN Algorithm for Pitch Detection**: For pitch detection, `librosa` integrates the YIN algorithm, known for its robustness in detecting fundamental frequencies within noisy and polyphonic audio. YIN is based

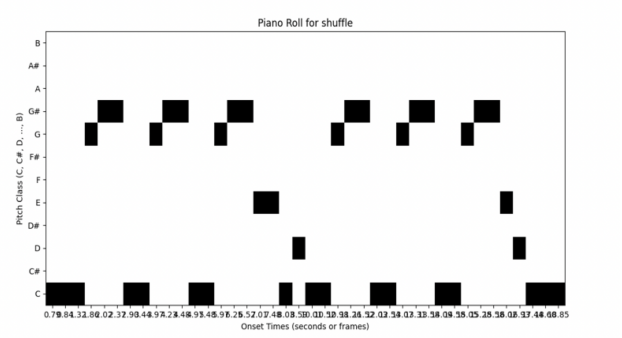


Figure 8. Piano Roll for Shufflin' Along (Librosa)

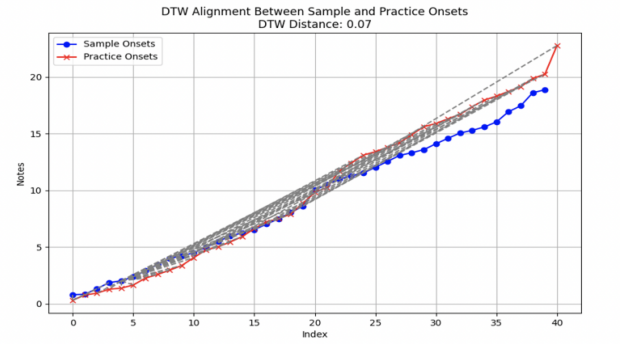


Figure 9. The analysis results for the first child's performance of Shufflin' Along.

on autocorrelation but improves upon traditional methods by introducing a difference function, which reduces the impact of harmonics and background noise, resulting in precise pitch estimates. This enhancement allows YIN to isolate the true pitch, making it particularly suitable for complex musical signals. Compared to the McLeod Pitch Method (MPM) and the Autocorrelation Function (ACF), YIN has distinct advantages in handling harmonically rich, multi-voiced audio. While MPM is effective for clean, single-pitch signals and ACF provides stable results in simple, monophonic cases, YIN is better suited for piano music. Piano compositions often include a mix of low and high frequencies with overlapping harmonics. YIN's ability to handle these complex harmonic structures enables it to capture the intricate pitch variations typical in piano performances.

3.3 Final Result

In preliminary testing, our application demonstrated promising results. We sourced multiple professional samples online and recorded several beginner piano students, comparing their recordings to these samples. The results generated by the system were subsequently evaluated against analyses conducted by a piano expert. The initial test piece was Shufflin' Along.

The first child received a very positive evaluation from the teacher, and the machine-generated DTW distance was

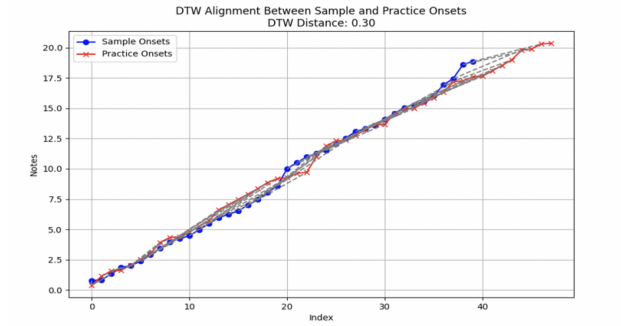


Figure 10. The analysis results for the second child's performance of Shufflin' Along.

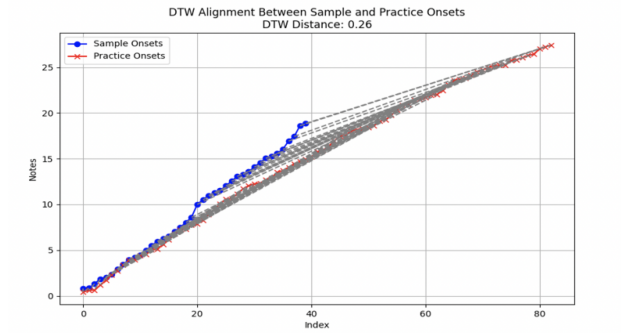


Figure 11. The analysis results for the third child's performance of Shufflin' Along.

only 0.07, indicating that their performance was very close to the sample (Figure 9).

The second child's performance was slightly less fluent, but they still played the basic melody correctly, with a final DTW distance of 0.3 (Figure 10).

The third child's performance also received favorable feedback from the teacher. Although there was background noise due to someone speaking during the recording and their playing tempo was significantly slower than the sample, the app performed well and was able to recognize the melody, with a DTW distance of 0.26 (Figure 11).

At the same time, we also tried uploading two completely different pieces of music (*Shufflin' Along* and *Pulling Out the Carrot*) for testing, and the DTW distance was significantly large, reaching 2.02 (Figure 12).

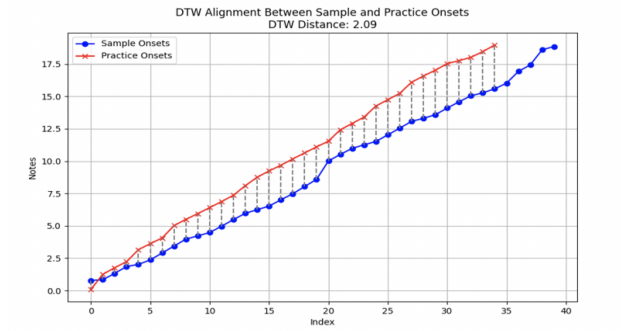


Figure 12. The analysis results for Different Piano Pieces.

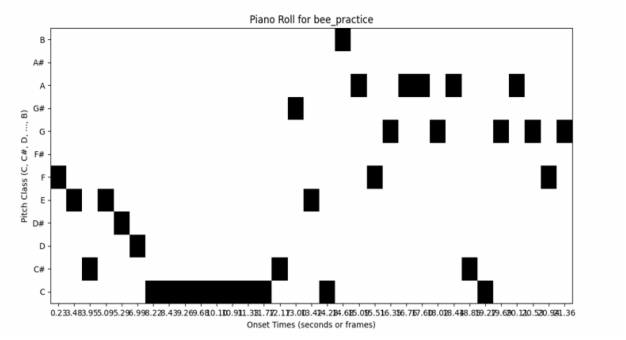


Figure 13. Piano Roll for Flight of the Bumblebee (practice)

Figure 14. Piano Roll for Flight of the Bumblebee (sample)

However, our current app still struggles to perform effectively with more complex and challenging pieces. For instance, Flight of the Bumblebee, known for its rapid tempo and intricate structure, resulted in distinctly different melodies being identified from the sample and the user recording (Figure 13-15).

4. DISCUSSION AND CONCLUSION

In this study, we developed a piano practice evaluation tool that utilizes onset detection, pitch detection, and a weighted Dynamic Time Warping (DTW) algorithm to assess beginner performances against professional samples. Our preliminary tests demonstrated the tool's effectiveness, as it accurately evaluated student recordings, even in cases with mild background noise or moderate tempo differences.

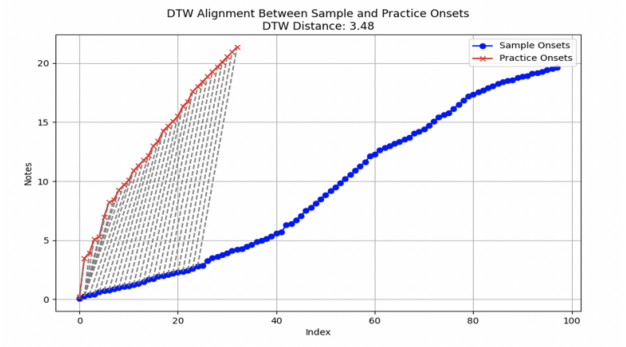


Figure 15. Flight of the Bumblebee - DTW

By applying a weighted DTW approach, we ensured that prominent notes held greater significance in the alignment, enhancing the relevance of our evaluations in a musical context. Leveraging librosa's onset and pitch detection algorithms, as well as the YIN algorithm, allowed us to achieve high accuracy in melody extraction, effectively capturing both timing and pitch information. For beginner pieces such as Shufflin' Along, the alignment results closely matched expert evaluations, confirming the tool's potential in real-world applications.

However, we encountered limitations when evaluating complex, high-speed compositions. For example, Flight of the Bumblebee, known for its intricate and rapid passages, led to discrepancies in detected melodies between the sample and student recordings. Such cases suggest challenges for our current algorithm in handling pieces with frequent pitch and intensity changes within short time frames.

Moving forward, we aim to address these challenges by incorporating several improvements. First, we will integrate advanced noise reduction techniques to enhance accuracy in noisy environments, making the tool more robust in practical settings. Additionally, we plan to embed more music theory into the algorithm, allowing for evaluations that consider rhythm, dynamics, and stylistic interpretation, which are critical in music education. Lastly, by exploring advanced machine learning models trained on larger, diverse datasets, we can improve the tool's ability to generalize across varied compositions and performance styles. Overall, these developments show promise for expanding the tool's real-time feedback capabilities in beginner piano practice and supporting a broader range of musical compositions.

5. REFERENCE

1. R. K. Sawyer, "Improvised conversations: Music, collaboration, and development," in Psychology of Music, vol. 27, 1999, pp. 192–216.

2. ABRSM, "Teaching, learning and playing music in the UK," Retrieved from https://gb.abrsm.org/en/making-music/4-the-statistics/, 2014.

3. CINIC, "Development status and market prospects of China's music education industry in 2016," Retrieved from http://www.chyxx.com/industry/201608/439530.htm 2016.

4. M. Mosing, G. Madison, N. Pederson, R. Kuja-Halkola, and F. Ullén, "Practice does not make perfect: No causal effect of music practice on music ability," Psychological Science, vol. 25, 2014, pp. 1795–1803.

5. L. Page, "Developmentally appropriate music practice: Children learn what they live," Young Children, vol. 56, no. 3, 2001, pp. 32–37.

6. K. McCord and E. H. Watts, "Collaboration and access for our children: Music educators and special educators together," Music Educators Journal, vol. 92, no. 4, 2006, pp. 26–33.

7. Zhang Weiwei, Chen Zhe, Yin Fuliang, et al. ”Review on Melody Extraction from Polyphonic Music” [J]. Acta Electronica Sinica, 2017, 45(4): 1000-1011. (in Chinese)

8. Li Sizhuan, Wang Hongcheng, Gu Siheng, Zhang Jiesen, Wan Yongqing. ”A Piano Note Onset Detection Algorithm Based on Fusion of Time-Frequency Information.” School of Information Science and Engineering, East China University of Science and Technology, Shanghai .

9. Gao Fei. ”Music Note Feature Recognition Based on the DTW Algorithm.” School of Music, Huai’nan Normal University, Huai’nan, Anhui 232001,2019.