# PianoES-A Piano Evaluation System For Learning

Ng Hui Ling
School of Computing, National University of Singapore
Singapore
e0695926@u.nus.edu

Nicholas Mak Hoi Hean
School of Computing, National University of Singapore
Singapore
nicholas.mak@u.nus.edu

Chen Yisheng Jonathan
School of Computing, National University of Singapore
Singapore
e0671506@u.nus.edu

Xing Wanting
School of Computing, National University of Singapore
Singapore
e0740939@u.nus.edu

## ABSTRACT

Learning to play the piano brings a multitude of benefits to learners, ranking from improvements in auditory sensitivity, speech skills, memory, diligence and focus, and creativity. However engaging a piano teacher to conduct piano lessons is expensive. Self-practice is typically difficult and ineffective without timely and accurate feedback. As such, a score-informed transcription and evaluation system, PianoES, is proposed to automatically convert the learners' piano recordings into score representations for comparison with the reference score to provide feedback on incorrect of missing notes. First, the onset timings of the piano recordings are aligned with the reference score using the Dynamic Time Warping (DTW) algorithm to generate a mapping function. Next, the recordings are transcribed into Musical Instrument Digital Interface (MIDI) representations using the ByteDance piano transcription model and aligned to the reference score using the obtained mapping function. Finally, a sequential algorithm is applied on the aligned MIDI representations to classify the incorrect or missing notes played by the learner. Our model is able to accurately evaluate the sequence of correct monophonic piano notes played, and promptly presents the feedback to piano learners for self-correction. The project page, which includes the demo application, datasets, and codes, can be viewed at: https://pianoes.github.io/.

## 1 INTRODUCTION

Learning to play the piano is beneficial for all ages. Playing the piano can improve auditory and speech skills for children and promote cognitive reserve for adults [14]. Despite the multitude of positive effects, the high cost of piano lessons may deter learners from starting or developing further musical skills [1], and it is not easy for learners to practise effectively on their own without timely feedback. Furthermore, the quality of piano teachers vary, and not all teachers may be able to accurately sieve out minute mistakes. The best teachers will also have their schedule packed, and would only be able to effectively teach a limited number of students at any given time.

### 1.1 Motivation

There is currently limited research focusing on score-informed transcription and evaluation systems for piano evaluation. These systems can play an important role in alleviating the costs of piano learning, encouraging learners to self-practice effectively. Such a system would be able to automatically transcribe piano recordings into score representations and align to the reference score for comparison to feedback misplayed notes to the learners.

### 1.2 Problem Statement

The following are three main challenges in developing an effective piano evaluation system.

- **Score-Audio Alignment** The onset timings of the recorded learners' and the reference audio must be aligned accurately to determine the correct rhythm. The challenge is to balance between robustness and accuracy after alignment [5].
- **Piano Pitch and Onset Detection** The piano has 88 notes, and each note produces a spectrum of harmonics which varies from different pianos [3], which can be difficult to identify. The piano is also polyphonic where multiple notes can be played simultaneously, resulting in source separation problems [3].
- **Onset Classification** After aligning the onset timings and detecting the pitch played, the challenge is to classify the incorrect or missing notes in a sequential manner.

### 1.3 Outline of Solution

The solution will focus on building a robust scored-informed transcription system, PianoES, that can provide immediate feedback and effective evaluation on note correctness.

## 2 RELATED WORK

### 2.1 Scored-Informed Music Information Retrieval

Based on current knowledge, there are only two formal studies conducted in 2012 [2] and 2017 [16] that focus on score-informed transcription systems for piano evaluation. Both systems are generally conducted with the similar steps, which include reference score and recorded audio alignment, pitch detection, and score-informed onset classification.

Both studies use the recordings from the Associated Board of the Royal Schools of Music 2011/12 syllabus for grades 1 and 2 as their dataset. The more recent study modified the provided annotations as it seemed to be too strict since mistakes around close vicinity seems to be played correctly by hearing [16].

Both studies use chroma features as audio representation for score-audio alignment. Dynamic Time Warping (DTW) [16] and

Window Time Warping (WTW) [2] algorithms are used for score-audio alignment for the respective studies. Non-negative matrix factorisation is employed in both studies for multi-pitch detection. The onset notes are then classified into incorrect, missing, or extra notes. Incorrect notes are determined by different temporal tolerance in each study, within 200 ms [2] and 250 ms [16]. F1-score was determined using different formulas in the studies. One study uses weighted F1-Scores for all classification [2], while another uses average F1-Score for each note classification [16], hence the accuracy is used to compare between the studies. The earlier study [2] shows that the missing notes (accuracy of 49.2%) are the most challenging to detect, followed by extra notes (accuracy of 60.5%). Incorrect notes can be detected well (accuracy of 93.2%). It is noted that the more recent study [16] performs better with the accuracy of 89.9%, 64.0%, and 98.6% for missing, extra, and incorrect notes detection respectively, but it was noted that the annotations have been modified and the threshold was set with an additional of 50ms more temporal allowance than the earlier study. Interestingly, the extra notes detection was consistently not performing well for both studies.

To understand better on each of the methods under the score-informed transcription system, audio synchronisation and piano transcription are further explored in the following related works.

## 2.2 Audio Synchronisation

Audio synchronisation or audio alignment is the task of aligning two different audios based on timings [5]. It is done to enable comparison between the pair of audio sequences.

The most common and longstanding alignment algorithm is DTW. DTW finds alignment by determining the optimal path between the audio features pair using dynamic programming. There are many variations of DTW, which includes FastDTW [12], which can greatly reduce execution time, and On-Line Time Warping [4] which is useful for real-time systems. WTW is another method that divides the path into a series of DTW windows and is suitable for longer audio sequences and real-time applications [11].

For alignment algorithms to be robust, the choice of audio features representations are crucial as they will provide patterns and information for the optimal path. There are many innovative explorations in the use of different features to compare the robustness in alignment.

In an alignment study for a complex orchestra music using the DTW alignment algorithm, the chroma, MFCC, and modified MFCC features are extracted and compared respectively with a range of parameters, which includes FFT sizes, hop sizes, number of MFCC, number of coefficients for MFCC, and the euclidean, city block, and cosine distance measures [6]. The study shows that there is no clear pattern in tuning the best parameters [6]. However, the MFCC and modified MFCC achieve the best results [6].

Another study on piano, orchestra, and strings audio alignment compares the chroma onset (CO), their novel decaying locally adaptive normalised CO (DLNCO), and the combination of both CO and DLNCO, where the combination of features significantly outperforms the others in accuracy [5].

Overall, the chroma feature seems to be a popular representation for audio alignment across all studies.

## 2.3 Piano Transcription

There are many studies on polyphonic piano transcription due to the difficulty in separating the spectral component of polyphonic music [18]. Overall, multi-pitch estimation methods can be broadly classified into three categories: feature-based, statistical model based, and spectrogram factorisation-based methods [18].

Feature-based methods extract features from time frequency audio representation to estimate joint multiple pitches, while statistical model-based methods include Maximum A Posteriori (MAP) estimation to select the most salient pitch in each iteration [18] or non-parametric Bayesian models [16]. Spectrogram factorisation-based methods decompose the spectrogram mixture into linear combination notes with corresponding intensities or probabilities, and the model parameters are estimated using expectation maximisation or non-negative matrix factorisation (NMF).

On top of those, deep learning methods have also been proposed, such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) to predict onsets and pitches from spectrograms [16]. A recent method combines a Convolutional Recurrent Neural Network (CRNN) and Sequence-to-Sequence model with an attention mechanism for joint multi-pitch detection and score transcription [10]. However, deep learning methods require a large diversity of annotated dataset to obtain good performances. A study proposes modification in the cost function deep learning model to overcome data scarcity errors in deep learning models [17].

# 3 PROPOSED METHOD

Figure 1 shows an overview of the proposed score-informed transcription system (PianoES) for piano evaluation, which can be broadly classified into three main steps. First, the recorded audio from a player is aligned with the reference audio using DTW to obtain the mapping function for onset timings. Next, the recorded audio is transcribed into MIDI format to extract the pitch and onset timings using the ByteDance piano transcription model, and re-aligned with the reference MIDI using the mapping function. And thirdly, a sequential mapping algorithm is applied to both the recorded and reference MIDI formats to classify incorrect notes. To evaluate the system performance, a data collection plan is curated which consists of the reference (i.e. score for players to follow) and the query (i.e. recorded audio by players) dataset. The accuracy, precision, recall, and F1-Score are used to evaluate the performances. The data collection process is detailed under Section 4. The system will be further developed into a functional web application, with the details under Section 5.

## 3.1 Score-Audio Alignment

Score-audio alignment is an important first step to ensure that the onset timings between the reference and the query audio are comparable. The reference audio contains the exact onset timings to be matched, while the recorded query audio may contain a variation of timings that can be interpreted as different speed, rhythm, and tempo. Accounting for the playing variability is important as it enables the model to achieve consistent results for players of differing skill level, who might opt to play at different speed, rhythm, and tempo.

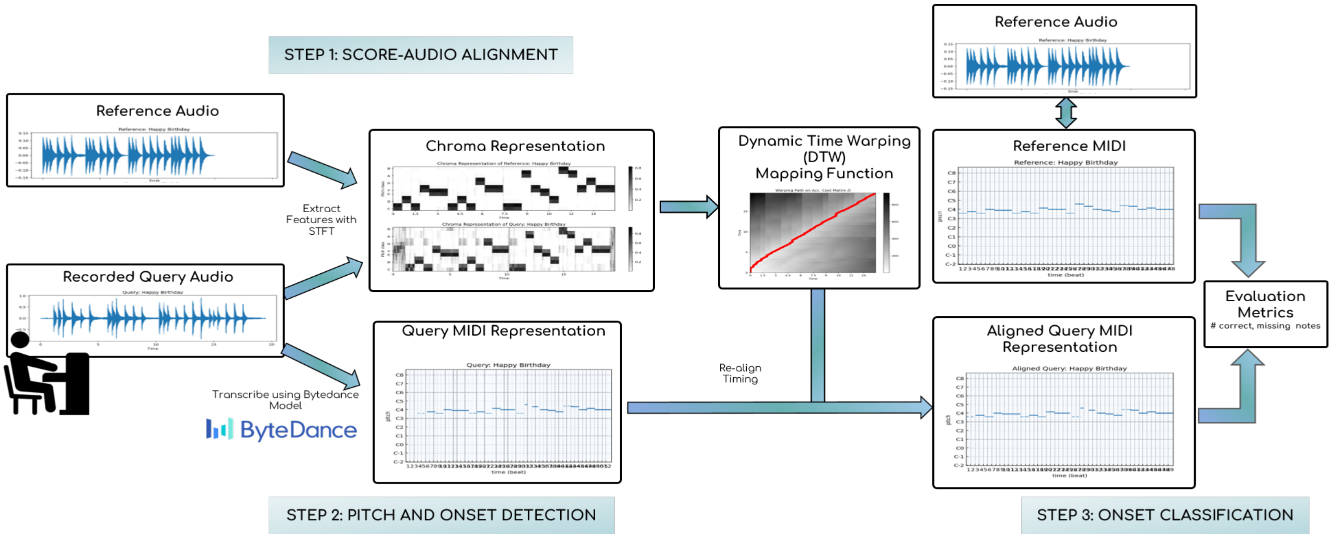
STEP 1: SCORE-AUDIO ALIGNMENT
Reference Audio
Reference: Happy Birthday
Recorded Query Audio
Query: Happy Birthday
Extract Features with STFT
Chroma Representation
Chroma Representation of Reference: Happy Birthday
Chroma Representation of Query: Happy Birthday
Dynamic Time Warping (DTW) Mapping Function
Reference Audio
Reference MIDI
Reference: Happy Birthday
Transcribe using Bytedance Model
ByteDance
Query MIDI Representation
Query: Happy Birthday
Re-align Timing
Aligned Query MIDI Representation
Aligned Query: Happy Birthday
Evaluation Metrics
# correct, missing notes
STEP 2: PITCH AND ONSET DETECTION
STEP 3: ONSET CLASSIFICATION

**Figure 1: Overview of the Proposed Score-Informed Transcription System**

3.1.1 ***Background***. To achieve the unified time domain, there needs to be a mapping from one time domain to the other. This problem can thus be defined as a temporal registration (matching) problem. Given real world noisy data, it is not possible to achieve perfect matching. As such, the temporal registration problem can be defined as an optimisation problem, whereby two input sequences, m and n, are matched to find a temporal correspondence function c. This mapping is non-linear given that tempo may vary, and should minimise an error metric, E, as follows: E = $\sum_{(i,j)\epsilon M} d(u_i, v_j)$.

In the equation, d represents some distance metric greater than 0, ui represents a point in query sequence m, and vj represents a point in reference sequence n. As with most optimization problems, some constraints have to be imposed to obtain a solution. In this case, three constraints are required. First, matching has to be monotonically increasing, as music is inherently sequential. A later point cannot be matched with an earlier point. As such, for any matching (i, j) and (i', j'), i' > i would imply that j' >= j. Second, (1, 1) and (m, n) have to be initialised as matching pairs. This constraint is required as the algorithm would need some boundary conditions as starting points, and the start and end points are the most logical. Third, every i would match one or more j. Every j would also match one or more i. This situation arises due to sampling: a single note might have multiple sampled points, resulting in the many-to-one mapping. An implication of many-to-one mapping is that all i and j should be matched.

3.1.2 ***Dynamic Time Warping (DTW)***. An algorithm widely used to solve the temporal registration problem is DTW, which solves the problem using a path based approach, finding a path P = p1, p2, … pL minimising the total distance $\sum_{(i,j)\epsilon M} d(u_i, v_j)$.

The path is found via dynamic programming, whereby i and j are iterated over, and the minimum distance path is found and memorised. The optimal path can be stored as a function mapping the query time domain to the reference time domain. The mapping function will then be used to re-aligned the query MIDI representation for comparison with the reference MIDI representation.

One of the limitations in DTW relates to its boundary conditions, whereby (1, 1) and (m, n) are matching pairs. While the dynamic programming algorithm is still able to find the optimal pairing, the results when implemented are not ideal when the player's recording has silent periods at the start and end points. This is due to the representation of the optimal path as a function; having silent parts at the start and end would result in a skewed mapping function. Moreover, the initialisation constraint of DTW is violated, which has been shown to disproportionately affect the resultant mapping [15].

To overcome this limitation, the DTW implementation includes functionality to trim silent regions from the front and end of audio recordings. While the resultant functionality is improved, it relies on a certain threshold to determine whether an area is actually silent. This threshold is a hyperparameter that requires tuning, and might not accurately account for scenarios where a recording might just have poorer quality. Nevertheless, the current tuning is sufficient for most scenarios.

3.1.3 ***Chroma Representation***. To ensure robust and accurate alignment, the audio pairs need to be extracted into a feature representation to provide patterns and information for the alignment. Studies have experimented with different features for alignment in the various domains. Although MFCC and modified MFCC seem to perform better than chroma for orchestra audio alignment [6], the chroma feature representation has been very popular, especially in piano audio alignment [2, 16]. Furthermore, the current studies on score-informed transcription systems use chroma feature representation for piano audio alignment, which has also proven high performances [2, 16]. Hence, chroma representations will also be proposed for this system for the audio alignments.

## 3.2 Pitch and Onset Detection

Score-informed representation (such as pitch and onset timings) has to be further extracted from the recorded query audio for downstream classification tasks.

*3.2.1* ***Background***. The polyphonic and harmonic nature of the piano results in challenges to extract accurate pitch and onset information. The polyphonic notes will aggregate into a waveform mixture and create challenges in source separation, and the harmonics could result in extra pitches being identified. Currently, deep learning methods have provided breakthroughs in state-of-the-art performances in multi-pitch detection due to the innovative model structure and the availability of large sets of piano data, including polyphonic chords, that are used for training.

Instead of reinventing another deep learning piano transcription model for our system, exploration is done on the various open-source deep learning piano transcription models, to determine the best model to be used in our system. The exploration reveals that the ByteDance model is one of the most recent developments and produces state of the art performance across the other deep learning models. Hence, the ByteDance model will be proposed for our score-informed piano transcription.

*3.2.2* ***ByteDance Piano Transcription Model***. The ByteDance piano transcription model provides a high-resolution piano transcription system by regressing velocities, onsets, offsets and classifying frames.

*Model Structure.* The ByteDance model [9] contains a velocity regression submodule, an onset regression submodule, a frame-wise classification submodule and an offset regression submodule (Fig 2). The velocity information of a piano note can be helpful to detect its corresponding onset, and the detection of onsets and velocities can affect each other. Therefore, the prediction of velocities is used to condition the prediction of onsets. The model concatenates the outputs of the velocity regression submodule and the onset regression submodule along their piano note dimension, and uses this concatenation as input to a biGRU layer to calculate the final onset predictions. Similarly, the model concatenates the outputs of the onset regression and offset regression submodules, and uses this concatenation as the input to a biGRU layer to calculate the final frame-wise predictions [9].

*Model Performance.* The ByteDance model was trained for 200k iterations using the MAESTRO dataset. Table 1 illustrates a comparison between precision, recall and F1 scores of the Bytedance model, the adversarial onset and frame model as well as the music transformer model. All models are evaluated using the evaluation dataset of MAESTRO. Bytedance model generally has the best F1 score across the frame, note and note with offset.

A series of experiments was performed using the ByteDance model to achieve a deeper understanding of the model algorithm as well as its performance. The ByteDance model's performance was evaluated on monophonic piano playing using the Piano dataset, which contains 127 audios from the 6 first exercises of Hanon's The Virtuoso Pianist. Referring to Table 2, the model takes an average of 7 seconds to complete the transcription of each audio. 20 piano recordings of Chopin's Nocturne with average lengths of 331s were used to further test the model's execution speed when handling recordings with higher complexity(polyphonic recordings), and an average evaluation time of 18s was achieved by the ByteDance Model. Therefore, because of the high F1 score achieved, and the efficient evaluation speed when handling both simple monophonic piano recordings and complex polyphonic piano recordings, the ByteDance Model was implemented in our system for piano transcription.

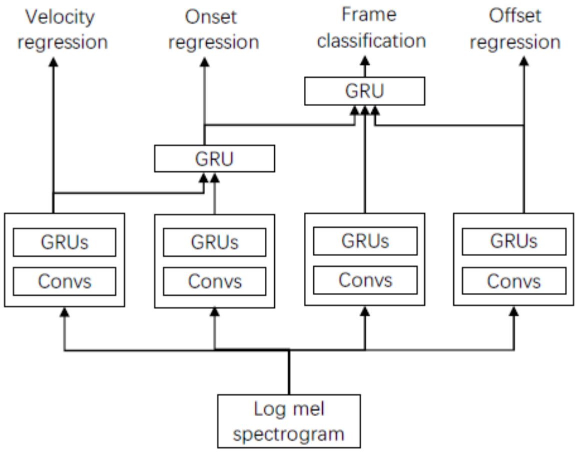

**Figure 2: ByteDance Transcription Model Structure [9]**

## 3.3 Onset Classification

After the transcription of the query audio into MIDI representation and time aligned with the reference MIDI representation, a comparison can then be made to classify the notes for evaluation.

*3.3.1* ***Background***. Based on the studies on the score-informed piano transcription system, the compared notes are classified into incorrect, missing, and extra notes. While incorrect notes can be detected with high performances (accuracy 93.2% [2] and 98.6% [16]), the missing notes have inconsistent performance (accuracy 49.2% [2] and 89.9% [16]) the extra notes are consistently difficult to detect in both studies (accuracy 60.5% [2] and 64.0% [16]). As such, our system will be focusing on detecting incorrect notes and missing notes.

One of the main challenges is the accumulated errors from time alignment or inaccurate pitch detection from the previous methods, which could further result in less accurate onset classification. To cater for time alignment errors, a range of temporal tolerance thresholds starting from 250 ms [16] will be implemented in our system, and another pure comparison without time alignment will also be studied and compared (further discussed in under sections 3.3.2 and 3.3.3). It is also crucial to design the sequential mapping algorithms that can classify notes accurately. Other challenges include the difficulty in identifying misplayed notes within close proximity with each other [16], which will be taken into consideration during the data collection process (under section 4.1).

| 2*Pitch-Detection Model | Frame | | | Note | | | Note with offset | | |
|---|---|---|---|---|---|---|---|---|---|
| | P% | R% | F1% | P% | R% | F1% | P% | R% | F1% |
| **ByteDance Model** | **88.71** | **90.73** | 89.62 | **98.17** | **95.35** | 96.72 | **83.68** | **81.32** | 82.47 |
| **ByteDance Model (Noise Labels)** | **84.65** | **91.36** | 87.79 | **98.65** | **94.30** | 96.39 | **80.59** | **77.09** | 78.77 |
| **Adversarial Onsets and Frames Model** | **93.10** | **89.80** | **91.40** | **98.10** | **93.20** | **95.60** | **83.50** | **79.30** | **81.30** |
| **Music Transformer Trained on MAESTRO** | **93.10** | **85.76** | **89.19** | **97.42** | **92.37** | **94.80** | **81.84** | **77.66** | **79.67** |

**Table 1: Evaluation Performance of Transcription Models [7–9]**

| DataSet | Longest Audio Length/s | Average Audio Length/s | Longest Evaluation Time/s | Average Evaluation Time/s |
|---|---|---|---|---|
| Piano Dataset | 32 | 18 | 11 | 7 |
| Chopin Nocturne Recordings | 368 | 331 | 26 | 18 |

**Table 2: ByteDance Model Efficiency Performance**

#### 3.3.2 *Combination of DTW and Sequential Mapping*.

For the first classification algorithm, DTW will be applied to align both the reference and query samples. The rationale for using DTW alignment before evaluation is two-fold. First, the query sample is very likely to be at a different tempo, and therefore time scale, as compared to the reference files used for evaluation. Without using a temporal alignment, it would be difficult to evaluate the user's rhythm, which is temporal dependent, as the timing of each note occurrence would appear different to the evaluation algorithm. Second, DTW is envisaged to simplify note matching and improve robustness of the evaluation algorithm. By aligning both reference and query temporally, the algorithm would, in theory, only need to search within a small temporal threshold to determine whether a note has been played, and whether it was played correctly. Moreover, with the definition of a temporal threshold, the evaluation algorithm would also be able to handle any additional noise notes (be it due to user error, or the piano transcription model), as it would already know what notes are expected to be present at each point in time.

Following DTW alignment, a sequential matching algorithm would be used to count the number of correct and missed notes played by the user. Sequential matching is used as music is inherently sequential; the notes have to be played in a particular order. To further enhance the model's robustness against noise points in the query, the algorithm would iterate over the reference notes, checking against the query notes at the same point in time (within a threshold) to determine whether a note was played correctly.

#### 3.3.3 *Sequential Mapping*.

Due to the limitations of DTW (under section 3.1.2), a second classification algorithm comprising only sequential mapping was evaluated. This algorithm uses a sliding window approach: for every reference note, a window of query notes is checked to determine whether the reference note was played. This window slides down the sequence of query notes once a match has been made. Due to the lack of any temporal checking, this algorithm can only determine whether each reference note was played; evaluation of rhythm is not possible.

Using a sliding window approach is robust to noise up to a certain point depending on the window size, as each reference note is checked against multiple notes in the window. However, care has to be taken to limit the size of the window: a larger window size increases the chances of a false positive matching, which would result in false, out-of-sync pairings for the remaining notes.

In essence, window size is a hyper-parameter that requires tuning. One possible method for tuning could be to link the window size to a difficulty level which the user selects. A higher difficulty level would correspond to a smaller window size, thereby leaving less room for errors and noise on the user's end.

## 4 EXPERIMENTS

### 4.1 Dataset

In order to evaluate system performance, the reference (i.e. scores for learners to follow) and query (i.e. recorded audio played by learners) dataset were collected.

#### 4.1.1 *Reference Dataset*.

Ten reference scores and MIDI files were created using ABC notation from MusPy, a toolkit for symbolic music generation. The reference scores were created with the constraints of having 4/4 time signature, in C Major, and at the pace of the 140 beats per minute. The number of notes varies for each reference score. Reference 7, 8, 9 contains polyphonics notes. Table 3 shows an overview of the reference dataset.

| Ref | Title | Total No. of Notes | Contains Polyphonic Notes? |
|---|---|---|---|
| 1 | Hot Cross Buns | 13 | No |
| 2 | Happy Birthday | 25 | No |
| 3 | Twinkle Twinkle Little Stars | 42 | No |
| 4 | Mary Had A Little Lamb | 26 | No |
| 5 | Notes Progression | 25 | No |
| 6 | Sonata No. 8, Pathetique, 2nd Movement | 23 | No |
| 7 | The Entertainer | 77 | No |
| 8 | Someone Like You | 44 | Yes |
| 9 | To Zanarkand | 81 | Yes |
| 10 | Edelweiss | 70 | Yes |

**Table 3: Overview of Reference Dataset**

*4.1.2* ***Query Dataset.*** There are five query test sets for each reference score, as shown in Table 4. Test Set 1 contains all the correct notes, Test Set 2, 3, 4 contains approximately 10% of the incorrect, missing, extra notes respectively, based on its respective total number of notes in the reference score. Test Set 5 contains minimally an incorrect, missing, extra note or approximately 10% of the total notes splitted equally into an incorrect, missing, extra note. Incorrect notes refer to the note that is played in different pitches, missing note refers to unplayed note, and extra note refers to additional note played. The exact rhythm is not annotated and accounted for.

Each test set of the respective reference score is manually annotated accordingly as close to realistic and practical considerations, where mistakes notes are not placed too near with each as it can be difficult to decipher by ear, to ensure proper evaluation. Instead of synthesising the annotated piece using the computer, a pianist is employed to play on the pieces, to mimic as close to real human performances. To ensure consistency in the evaluation of all the reference pieces, the pieces were all played using a digital piano in a quiet but not sound-proof environment. They were recorded and saved in a m4a audio format. The playing environment aims to mimic actual practising environment, where noise can be interjected at times, to provide a more realistic evaluation on our system.

## 4.2 Metrics

Accuracy, precision, recall, and F1-score are used to evaluate the system performance. As accuracy is the main metric used to compare in the past research studies, it will be used for standard comparison in our system. The current standard system yields around an accuracy of more than 90% (93.2% [2] and 98.6% [16]) on incorrect notes, and ranges from the accuracy of 49.2% [2] and 89.9% [16] for missing notes detection.

## 4.3 Results

*4.3.1* ***ByteDance Piano Transcription Model.*** The ByteDance piano transcription model provides accurate pitch detection from the recorded query audio. However, there are some pieces which have additional pitches detected which could be caused by harmonics or environment noises, as shown in Fig 3 and 4, however this will not affect the onset classification later on, as sequential mapping uses reference MIDI as the basis for matching (i.e. match query to reference instead of reference to query). Other than the extra pitches, the ByteDance model can detect all the correct pitches, even in polyphonic notes, as shown in Fig H.

*4.3.2* ***Combination of DTW and Sequential Mapping.*** The combined algorithm of DTW alignment, and sequential matching, with reset mechanism was tested with the test set consisting of all correct notes (Test Set 1) across Reference 1 to 7, using a variation of temporal tolerance thresholds, starting at 250 ms. It should be taken in caution that correct notes in our system are denoted by notes that are played correctly in terms of correct pitch, but the onset timings might not be aligned with the reference timing due to human errors.

Since Test Set 1 is our ground truth correct notes dataset, it is ideally expected that all the test cases meet 100% accuracy at the lowest threshold of 250 ms. However, the results show otherwise, which infer that our ground truth dataset might not be played at the correct tempo. This is quite predicted as it is realistic that humans playing on the piano can go off beat or tempos, unlike a computer playing. Nevertheless, at the strictest threshold of 250 ms, the average accuracy for incorrect notes is around 80%, as shown in Table 5. To reach at least 90% average accuracy in our ground truth data, a threshold of 600 ms is required.

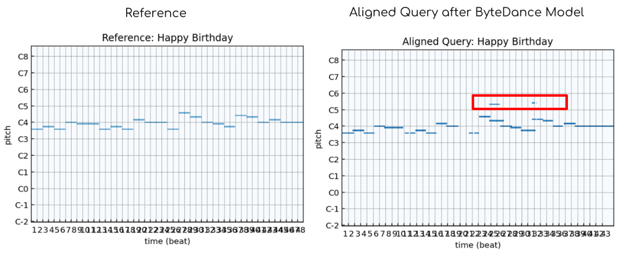

**Figure 3: Pitch Detection Errors from ByteDance Transcription Model (Monophonic Song)**

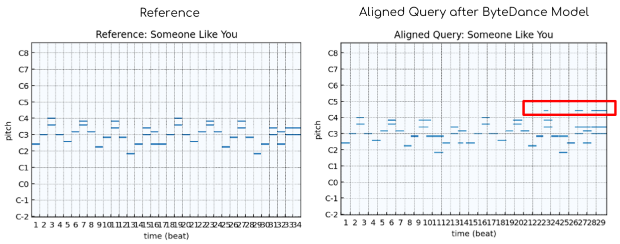

**Figure 4: Pitch Detection Errors from ByteDance Transcription Model (Polyphonic Song)**

As similar research studies used the thresholds of 200 ms and 250 ms to identify correct notes, an analysis was done at the low 250 ms thresholds in our system. It was tested that at the low threshold of 250 ms, it is difficult for the human to hear a difference in the incorrect notes (i.e. notes that are more than 250 ms from the reference onset timing) detected by our model. As such, it is difficult to tell if low thresholds are useful for piano evaluation, since the human cannot hear the difference and thus, it might be too strict to be implemented in a piano evaluation system for learning. Even at the 600 ms (0.6 seconds) threshold, it is still difficult to hear the difference and determine if it is indeed off rhythm.

However, using large thresholds seems to be problematic as well. From the results in Table 5, a very high threshold was needed in order to hit the expected accuracy of 100%, especially with the longer pieces (e.g. Reference Scores 6, 7). The high threshold required mainly comes from the inherent difficulty for the user to keep a constant tempo throughout a longer piece. Even with the DTW alignment and the reset mechanism, small errors in tempo accumulate and propagate down the piece. For example, the reference 7 test query lasted 42 seconds, even though it was only expected to last for 35 seconds at the same tempo. These errors proved to be too large for the DTW alignment to handle at the level

| Ref | Music Title | Test Set 1 (all correct) | Test Set 2 | Test Set 3 | Test Set 4 | Test Set 5 | | |
|---|---|---|---|---|---|---|---|---|
| | | | Wrong notes | Missing Notes | Extra Notes | Wrong notes | Missing Notes | Extra Notes |
| 1 | Hot Cross Buns | - | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | Happy Birthday | - | 3 | 3 | 3 | 1 | 1 | 1 |
| 3 | Twinkle Twinkle Little Stars | - | 4 | 4 | 4 | 1 | 1 | 1 |
| 4 | Mary Had A Little Lamb | - | 3 | 3 | 3 | 1 | 1 | 1 |
| 5 | Notes Progression | - | 3 | 3 | 3 | 1 | 1 | 1 |
| 6 | Sonate No. 8, Pathetique, 2nd Movement | - | 2 | 2 | 2 | 1 | 1 | 1 |
| 7 | The Entertainer | - | 8 | 8 | 8 | 3 | 3 | 3 |
| 8 | Someone Like You | - | 4 | 4 | 4 | 1 | 1 | 1 |
| 9 | To Zanarkand | - | 8 | 8 | 8 | 3 | 3 | 3 |
| 10 | Edelweiss | - | 7 | 7 | 7 | 2 | 2 | 2 |

**Table 4: Overview of Query Dataset**

of precision required for thorough and accurate analysis. Even if the DTW alignment was able to handle the large shift, it would cause severe distortion in the query, which also contributes to the high thresholds required for high accuracy.

*Limitation 1: Lack of Ground Truth on Rhythm or Tempo.* Currently, the incorrect annotation is denoted on whether the notes are played correctly or not. There is no annotation if the rhythm is correct even though there might be variations in the rhythms or tempos that cannot be differentiated by the human ear, but only by the model. Hence it is challenging to identify or validate if the model is in fact detecting the notes correctly or if it is indistinguishable by the human ear.

*Limitation 2: Discrete and Continuous Mapping.* From the results, it can be seen that alignment is a major issue when conducting score evaluation. Apart from the above-mentioned reason of propagating errors, there are other issues which contribute to the difficulty in alignment. One such issue is that of discrete versus continuous mapping. DTW produces a discrete mapping of temporal points between the reference and query, based on a predetermined step size. However, the query sample, being a continuous audio time series, requires a continuous mapping function to be aligned to the reference sample. This can be achieved by approximating the discrete mapping with a continuous function.

In practice, this continuous mapping function is accurate when all notes are present. The issue arises when the user misses out notes. In such a case, DTW produces a many-to-one mapping, where many points in the query sample (especially in regions of missing notes) map to one small silent point in the reference audio. The approximating continuous function would thus be distorted, resulting in imperfect alignment (Figure 5).

*Limitation 3: Silence Periods.* Another issue is fundamental to the DTW algorithm itself. As part of the formulation of DTW, one constraint imposed to make it is that the first and last points of both sequences must be paired. Such a constraint is violated when there are periods of silence at the start of a query, as is common when a user presses 'record' and sits down in preparation for playing. The result of violating this constraint is a many-to-one mapping (Figure 6), the effects of which have been discussed above.

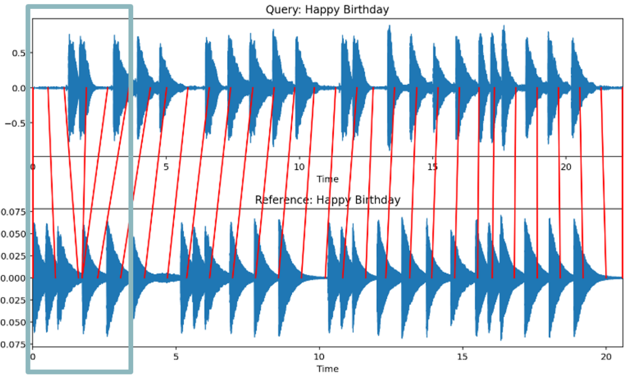

**Figure 5: Many-to-one mapping exhibited by DTW due to missing notes**

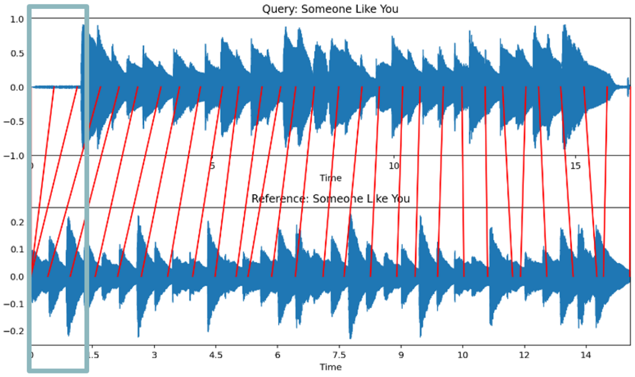

**Figure 6: Many-to-one mapping exhibited by DTW due to silence at start and end of recording**

| | % Correct Notes in Test Case 1 (all correct) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Threshold (ms) | Ref 1 | Ref 2 | Ref 3 | Ref 4 | Ref 5 | Ref 6 | Ref 7 | Ave |
| 250 | 84.6 | 76.0 | 81.0 | 84.6 | 80.0 | 78.3 | 77.9 | 80.3 |
| 300 | 84.6 | 80.0 | 85.7 | 88.5 | 84.0 | 82.6 | 80.5 | 83.7 |
| 400 | 92.3 | 84.0 | 88.1 | 88.5 | 88.0 | 87.0 | 84.4 | 87.5 |
| 500 | 92.3 | 88.0 | 90.5 | 92.3 | 88.0 | 87.0 | 83.1 | 88.7 |
| 600 | 92.3 | 88.0 | 90.5 | 92.3 | 92.0 | 91.3 | 85.7 | 90.3 |
| 700 | 92.3 | 88.0 | 92.9 | 92.3 | 92.0 | 91.3 | 87.0 | 90.8 |
| 800 | 100.0 | 92.0 | 92.9 | 96.2 | 92.0 | 91.3 | 90.9 | 93.6 |
| 900 | 100.0 | 92.0 | 95.2 | 96.2 | 96.0 | 91.3 | 92.2 | 94.7 |
| 1,000 | 100.0 | 92.0 | 95.2 | 96.2 | 96.0 | 95.7 | 92.2 | 95.3 |
| 2,000 | 100.0 | 96.0 | 97.6 | 100.0 | 100.0 | 100.0 | 96.1 | 98.5 |
| 3,000 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 97.4 | 99.6 |
| 8,000 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |

**Table 5: DTW Accuracy Across Thresholds for Test Set 1 (Reference 1 to 7)**

Attempts were made to circumvent this issue by clipping the recorded query audio using a volume threshold. This proved to be successful, but did not solve the underlying issue of representing a discrete many-to-one mapping as a continuous mapping function.

Due to the limitations and issues presented by the DTW algorithm, an alternative variant of the algorithm was tested. This new algorithm uses only sequential matching, without the aforementioned DTW aspects.

### 4.3.3 *Sequential Mapping.*

Generally, the system accuracy is reasonably high (Table 6). Taking an average across all test sets and reference material, our system achieves an accuracy of 84.4%. While not as high as the other systems described in Section 4.2, where the accuracy is more than 90%, it is believed that hyper-parameter tuning can further improve the performance.

One particular noteworthy result is that of reference 6. The accuracy for test 1 and 3 with this reference (all correct notes and missing notes respectively) was 100%, yet the accuracy for test set 2 was only 26.1%. This abnormally high error rate was found to have come from a phantom note arising from the piano transcription model, which pushed the correct query note out of the window. As a result, all the reference notes could not find an appropriate match, resulting in a 'false' classification for the entire song, and a corresponding low accuracy. This underscores the need for a more robust sliding window technique which can account for different levels of error, or perhaps even a different algorithm altogether.

The longer pieces (references 6 and 7) warrant further discussion and analysis, as they are close to what this system will encounter when eventually deployed. In reference 7, it can be seen that the performance for test sets 1 and 2 (correct notes, wrong notes respectively) was much worse than that of test set 3 (missing notes), in terms of accuracy and recall. Logically, the performance for test sets 2 and 3 should be similar, as sequential matching should be invariant to whether there are added or missing notes, with both cases being different sides of the same coin metaphorically. As such, the huge difference in performance can be attributed to the sliding window approach used - this has been explained in the preceding paragraph.

The sliding window approach can be thought of as a greedy algorithm, as it selects the best possible matching between notes within a small search space. For better performance, it might be prudent to test other algorithms which are more globally optimal. One class of algorithms that come to mind are the solutions associated with sub-string and sub-sequence pairing. Most of the solutions involve dynamic programming, which is usually able to obtain a globally optimal pairing, and by extension better results if used in this model.

# 5 WEB APPLICATION

In order to evaluate the learners' performance against a reference score, a web application was developed to provide a user-facing interface to test-bed the project.

## 5.1 Understanding Users

The target users of our application are mainly novice piano learners who have access to devices such as home PCs or tablets. The barrier to entry for these users should remain low, since the users focus is on learning to play the piano effectively, and not to learn how to use the app. With this in mind, the application interface was built to be simple and straightforward to use. Navigation is kept to minimal with only two intuitive screens for sheet selection and piano evaluation. This interface is also similar to other sight reading applications, such as PiaNote [13]. An initial design was done using Figma to test the usability of the interface design. It can be viewed here: PianoES Prototype.

The prototype was further iterated to consider the colouring of the annotation of notes on the sheet music so that the users can identify mistakes easily at a glance.

## 5.2 Web Application Features

### 5.2.1 *Variety of Songs.*

A variety of 10 classic piano songs were chosen for their popularity and variation in complexity and length.

| Ref | Music Title | Test Set | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|---|
| 1 | Hot Cross Buns | 1 (Correct) | 100.0 | 100.0 | 100.0 | 1 |
| | | 2 (Wrong Notes) | 100.0 | 100.0 | 100.0 | 1 |
| | | 3 (Missing Notes) | 100.0 | 100.0 | 100.0 | 1 |
| 2 | Happy Birthday | 1 (Correct) | 60.0 | 100.0 | 60.0 | 0.75 |
| | | 2 (Wrong Notes) | 100.0 | 100.0 | 100.0 | 1 |
| | | 3 (Missing Notes) | 88.0 | 95.2 | 90.9 | 0.93 |
| 3 | Twinkle Twinkle Little Stars | 1 (Correct) | 100.0 | 100.0 | 100 | 1 |
| | | 2 (Wrong Notes) | 90.5 | 94.7 | 94.7 | 0.95 |
| | | 3 (Missing Notes) | 90.5 | 94.7 | 94.7 | 0.95 |
| 4 | Mary Had A Little Lamb | 1 (Correct) | 100.0 | 100.0 | 100.0 | 1 |
| | | 2 (Wrong Notes) | 84.6 | 91.3 | 91.3 | 0.91 |
| | | 3 (Missing Notes) | 84.6 | 91.3 | 91.3 | 0.91 |
| 5 | Notes Progression | 1 (Correct) | 100.0 | 100.0 | 100.0 | 1 |
| | | 2 (Wrong Notes) | 84.0 | 90.9 | 90.9 | 0.91 |
| | | 3 (Missing Notes) | 88.0 | 95.2 | 90.9 | 0.93 |
| 6 | Sonate No. 8, Pathetique, 2nd Movement | 1 (Correct) | 100.0 | 100.0 | 100.0 | 1 |
| | | 2 (Wrong Notes) | 26.1 | 100 | 19.0 | 0.32 |
| | | 3 (Missing Notes) | 100.0 | 100.0 | 100.0 | 1 |
| 7 | The Entertainer | 1 (Correct) | 41.6 | 100.0 | 41.6 | 0.59 |
| | | 2 (Wrong Notes) | 40.3 | 96.0 | 34.8 | 0.51 |
| | | 3 (Missing Notes) | 96.1 | 98.5 | 97.1 | 0.98 |

**Table 6: Sequential Mapping Evaluation Results**

This allows us to remain relevant to piano learners of different skill levels (Figure 7).

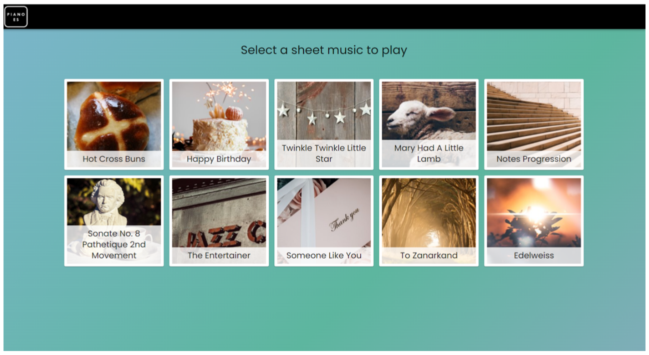
Select a sheet music to play

**Figure 7: Piano Song Options**

*5.2.2* ***Guidance to Learn Songs.*** The web application requires the browser to enable voice recording to record audio. The evaluation page also displays the sheet music in standard sheet notation form as well as an audio guide that users are able to listen to before commencing their practice (Figure 8).

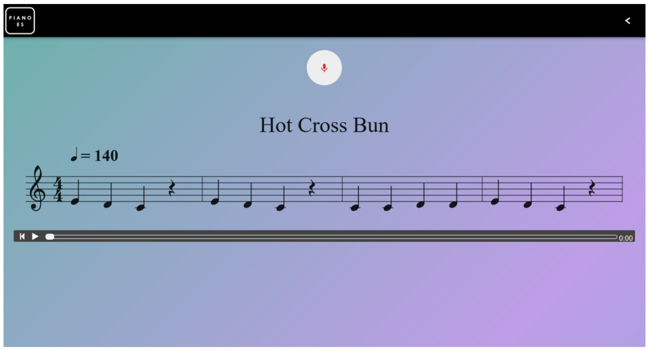

**Figure 8: Display of Piano Score**

*5.2.3* ***Robust Feedback on Recorded Audio.*** Playback recording is also available for learners to listen to their own recorded audio, which they can then download for further analysis if required. The evaluation system automatically highlights incorrect notes played on screen for visual feedback. In the full evaluation report, the number of hit and missed notes is also rendered on screen for learners' to understand their overall performance (Figure 9).

*5.2.4* ***Timely Feedback.*** The application is able to process and evaluate the audio recording in the backend within less than 20 seconds of the audio recording submission.

## 5.3 Web Application Design

A research on existing piano sight-reading applications was conducted prior to the design, with the focus on identifying useful tools and features to visualise music scores in the web. The visualisation was narrowed down to two most popular tools, abcjs and VexFlow. Ultimately, abcjs was selected, as it can be easily used create reference score as well. Other considerations are based on our expertise on the framework selected.

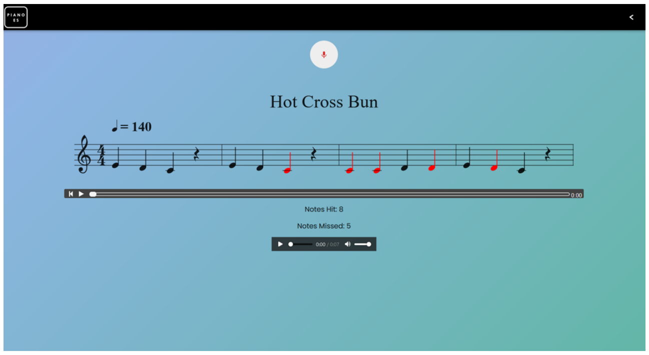

**Figure 9: Evaluation Results**

### 5.3.1 *Architecture*.

The web application is split into three main modules: (1) frontend using React.JS, (2) backend using Flask, and (3) a read-only database using Javascript Object Notation (JSON). The 3-layer architecture in Figure 10 was used to build a simple working prototype.

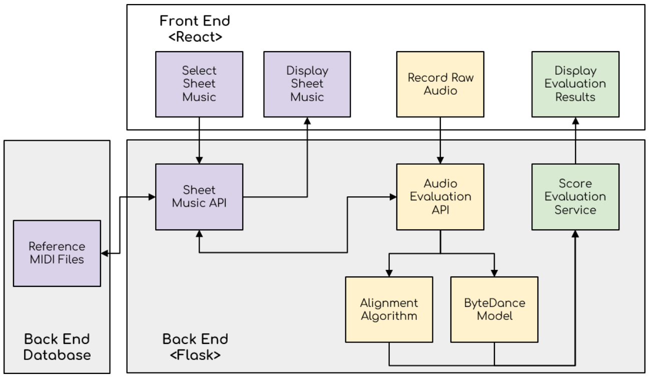

**Figure 10: Three layer architecture of frontend, backend and database**

### 5.3.2 *Deployment Server*.

The web application is deployed at https://pianoes.online using an AWS EC2 t3a.large server as the transcription model requires sufficiently large RAM in order to process the audio.

### 5.3.3 *Evaluation*.

The web application was tested against actual piano playing against multiple Hot Cross Buns piano playing and was found to be able to record the audio accurately as well as provide corrective feedback within 30 seconds (Table 7).

## 6 FUTURE WORK

A few areas for future optimizations of the PianoES system has been identified:

- **Incapability of handling complex user scenarios** Other than playing wrong notes or missing notes, users could also play extra notes or make rhythmic mistakes where query notes would be misaligned compared to the reference score. With the sequential matching algorithm that has been implemented in PianoES, identification of misaligned notes would not be supported. Also, current system evaluation and result analysis focus on monophonic music pieces only, while more advanced piano learners would require the system to be able to support polyphonic pieces as well. Therefore, note detection and alignment algorithms could be further improved in order to cover a wider range of user scenarios.
- **Phantom notes** Unexpected notes were detected in results generated from transcription models, which could be possibly caused by environmental noises. Given the wide possible range of user scenarios, being able to detect noises and separate them from the actual playing would be essential to maintain a stable system performance, and this would be an important feature that can be worked on in the future.
- **Pianos do not produce identical sounds** Different types of pianos would produce sounds that are different in colours and tones. Also, pianos might be tuned with different frequencies, making it challenging for the system to provide accurate evaluations of users' playing. This would be a possible field that PianoES can be improved in the future.
- **Unstable performance in different environment** It has been observed that when evaluation tasks are executed in different environments/on different computers, there could be significant differences in execution time. Therefore, the system could be further improved to ensure that it could support fast evaluation in most user scenarios.
- **High memory consumption** Execution of evaluation tasks consume very significant amount of computer memory, which would add unnecessary hardware restrictions to potential users and is in contrary to the motivation of building an easy-to-use application for piano learners. Therefore, future optimization in this area could be essential if the application would be promoted to a larger user base.

| Sample | Duration | Accuracy |
|---|---|---|
| 1 | 23.92 | 100.0% |
| 2 | 24.35 | 84.6% |
| 3 | 24.58 | 92.3% |
| 4 | 24.19 | 100.0% |
| 5 | 24.18 | 100.0% |

**Table 7: Repeated live playing of Hot Cross Buns and accuracy of evaluation report from model**

## 7 CONCLUSION

In conclusion, the score-informed transcription and evaluation system PianoES serves as an effective approach for automatic piano transcription. It serves as a practical solution for piano learners, especially piano beginners to carry out self-learning and self-practicing efficiently.

Various key concepts including score-informed music information retrieval, audio synchronisation and piano transcription have been studies and discussed during PianoES's development and implementation. Score-audio alignment, pitch detection and evaluation models have been experimented and carefully adopted in the system to achieve a comprehensive solution of identified problems.

Based on system evaluation results, PianoES is able to provide accurate and timely feedback to piano learners, hence effectively assist with their self-learning and evaluations.

Several limitations have also been identified, such as the problems with DTW alignment, and the lack of robustness when using a greedy sequential approach to match a reference and query audio for evaluation. Tackling these limitations would enable the system to be more robust to different user scenarios, and provide evaluation feedback to users with higher efficiency and accuracy in the future.

## 8 CONTRIBUTIONS

Table 8 indicates each group member's contributions:

| Member | Tasks |
|---|---|
| Nicholas | Data Collection<br>Modelling - Alignment<br>Modelling - Evaluation |
| Wanting | Modelling - Pitch Detection |
| Jonathan | Frontend<br>App Deployment |
| Hui Ling | Data Collection<br>UI Design<br>Backend<br>Modelling |

**Table 8: Group Member Contributions**

All group members contributed during preparation of the mid-project presentation, the final project presentation and the final report.

## REFERENCES

[1] ABRSM (2014 [Online]). Teaching, learning and playing in the uk.

[2] Benetos, E., Klapuri, A., and Dixon, S. (2012). Score-informed transcription for automatic piano tutoring. In *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 2153–2157. IEEE.

[3] Berg-Kirkpatrick, T., Andreas, J., and Klein, D. (2014). Unsupervised transcription of piano music. *Advances in neural information processing systems*, 27.

[4] Dixon, S. (2005). Live tracking of musical performances using on-line time warping. In *Proceedings of the 8th International Conference on Digital Audio Effects*, volume 92, page 97. Citeseer.

[5] Ewert, S., Muller, M., and Grosche, P. (2009). High resolution audio synchronization using chroma onset features. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1869–1872. IEEE.

[6] Gadermaier, T. and Widmer, G. (2019). A study of annotation and alignment accuracy for performance comparison in complex orchestral music. *arXiv preprint arXiv:1910.07394.*

[7] Hawthorne, C., Stasyuk, A., Roberts, A., Simon, I., Huang, C.-Z. A., Dieleman, S., Elsen, E., Engel, J., and Eck, D. (2018). Enabling factorized piano music modeling and generation with the maestro dataset. *arXiv preprint arXiv:1810.12247.*

[8] Kim, J. W. and Bello, J. P. (2019). Adversarial learning for improved onsets and frames music transcription. *arXiv preprint arXiv:1906.08512.*

[9] Kong, Q., Li, B., Song, X., Wan, Y., and Wang, Y. (2021). High-resolution piano transcription with pedals by regressing onset and offset times. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3707–3717.

[10] Liu, L., Morfi, V., and Benetos, E. (2021). Joint multi-pitch detection and score transcription for polyphonic piano music. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 281–285. IEEE.

[11] Macrae, R. and Dixon, S. (2010). Accurate real-time windowed time warping. In *ISMIR*, pages 423–428.

[12] Salvador, S. and Chan, P. (2007). Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580.

[13] Schulz, D. (2016). Pianote: a sight-reading program that algorithmically generates music based on human performance.

[14] Seinfeld, S., Figueroa, H., Ortiz-Gil, J., and Sanchez-Vives, M. V. (2013). Effects of music learning and piano practice on cognitive function, mood and quality of life in older adults. *Frontiers in Psychology*, 4:810.

[15] Silva, D. F., Batista, G., Keogh, E., et al. (2016). On the effect of endpoints on dynamic time warping. *SIGKDD MiLeTS*, 16:10.

[16] Wang, S., Ewert, S., and Dixon, S. (2017). Identifying missing and extra notes in piano recordings using score-informed dictionary learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(10):1877–1889.

[17] Weissman, R. (2021). Musical note recognition algorithms need tuning too.

[18] Zhang, W., Chen, Z., and Yin, F. (2020). Multi-pitch estimation of polyphonic music based on pseudo two-dimensional spectrum. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2095–2108.