ABSTRACT

There are many benefits in learning the piano, however it is expensive to take up piano lessons and difficult to self-practice effectively without timely and accurate feedback. As such, a score-informed transcription system, PianoES, is proposed to automatically convert the learners' piano recordings into score representations for comparison with the reference score to provide feedback on misplayed notes. First, the onset timings of the piano recordings are aligned with the reference score using the dynamic time warping (DTW) algorithm to provide a mapping function. Next, the recordings are transcribed into Musical Instrument Digital Interface (MIDI) representations using the Bytedance piano transcription model and realigned to the reference score using the obtained mapping function. Finally, a sequential algorithm is applied on the aligned MIDI representations to classify the incorrect notes played by the learner. <Explain summary of the results>
View our project page: https://pianoes.github.io/.

1. INTRODUCTION

Learning the piano is beneficial for all ages as it can improve auditory and speech skills for children and promote cognitive reserve for adults [1]. However, despite the multitude of benefits, the high cost of piano lessons may deter learners from starting or pursuing further [2]. It is also difficult for learners to practice effectively on their own without timely feedback to correct their playing. Furthermore, piano teachers might not be able to accurately sieve out all mistakes, and different teachers have their own method of teaching.

1.1 Motivation

There is currently limited research focusing on score-informed transcription systems for piano evaluation, which can play an important role in alleviating the high costs of piano learning and encouraging learners to effective self-practices. The system can 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.

(1) Detecting Rhythm
The onset timings of the recorded learners' and the reference audio must be aligned accurately to determine the correct rhythm. <add challenges>

(2) Detecting Piano Pitch and Onset
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].

(3) Classifying Misplayed Notes Sequentially
After aligning the onset timings and detecting the pitch played, it is challenging to extract the misplayed notes in a sequential manner. <add challenges>

1.3 Outline of Solution

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

1.4 Division of Work

<add workload>

2. RELATED WORKS

2.1 Audio Data Representation

There are different types of audio representation inputs

2.2 Score Data Representation

2.3 Piano Transcription

There are many research methods on polyphonic piano transcription as the spectral part of polyphonic music will affect each other [5]. Overall, multi-pitch estimation methods can be broadly classified into three categories, feature-based, statistical model based, and spectrogram factorisation-based methods [5].

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 [5] or non-parametric Bayesian models [6]. 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 maximization or non-negative matrix factorisation (NMF).

On top of those, deep learning methods have also been proposed such as Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) to predict onsets and pitches from spectrograms [6]. A recent method combines and uses Convolutional Recurrent Neural Network (CRNN) and Sequence-to-Sequence model with attention mechanism for joint multi-pitch detection and score transcription [4]. However, deep learning methods require a large diversity of annotated dataset to obtain good performances. Interestingly, a popular piano training application, Simply Piano, uses deep learning [7], however their collected dataset is limited and imbalanced, hence they proposed modification in their model's cost function to overcome data scarcity errors [8].

2.4 Scored-Informed Music Information Retrieval <can be refined contents>

There is some research which compares a learner's piano audio input with a reference score to detect errors and provide feedback to the learner. The piano playing audio of the learner is recorded and aligned with the reference MIDI score using alignment methods such as the windowed time warping (WTW) alignment algorithm, which has high performance and is computationally inexpensive [9] or using chroma with onset indicator features [10]. After the alignment,
the MIDI files were synthesised and transcribed using the NMF algorithms into piano-roll comparison [9].

3. PROPOSED METHOD

<write an overview here, add pictures of the system>

3.1 Score-Audio Alignment

Certain evaluation metrics, such as note accuracy, require a basis of comparison between the player's recording against a reference. For the comparison to be fair and robust, a unified time domain would have to be created; such a mechanism would account for variability in the tempo of different players. Accounting for tempo is important as it enables the model to achieve consistent results for players of differing

skill level, who might opt to play at different tempos. In such a case, metrics such as accuracy should give consistent results.

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:

$$E = \sum_{(i,j) \in M} d(u_i, v_j)$$

Where d represents some distance metric greater than 0, u_i represents a point in query sequence m, and v_j 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. This constraint, while required to solve the optimisation problem, led to several problems downstream, which will be discussed in Section xx.

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.

An algorithm widely used to solve the temporal registration problem is Dynamic Time Warping (DTW). DTW solves the problem using a path based approach, finding a path P = {p1, p2, … pL} minimizing the total distance $\sum_{(i,j) \in M} d(u_i, v_j)$. The path is found using a dynamic programming approach, whereby i and j are iterated over, and the minimum distance path is found and memoised. The optimal path can be stored as a function mapping the query time domain to the reference time domain.

The first limitation of 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 (https://core.ac.uk/download/pdf/78275924.pdf).

To overcome this limitation, our 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, we believe that our current tuning is sufficient for most scenarios.

3.2 Pitch and Onset Detection

<add figure number and link to text, add reference>
The ByteDance piano transcription model provides a high-resolution piano transcription system by regressing velocities, onsets, offsets and classifying frames.

3.2.1 Model Structure

Referring to [model structure figure to be added], this system contains a velocity regression submodule, an onset regression submodule, a frame-wise classification submodule and an offset regression submodule. *Each submodule is modeled by an acoustic model. Each acoustic model is modeled with several convolutional layers followed by bidirectional gated recurrent units (biGRU) layers. The convolutional layers are used to extract high-level information from the log mel spectrogram, and the biGRU layers are used to summarize long time information of the log mel spectrogram. Then, a time-distributed fully connected layer is applied after the biGRU layer to predict the regression or classification result for each pitch along the time axis<this part can be removed if need space for more important content>*.

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.

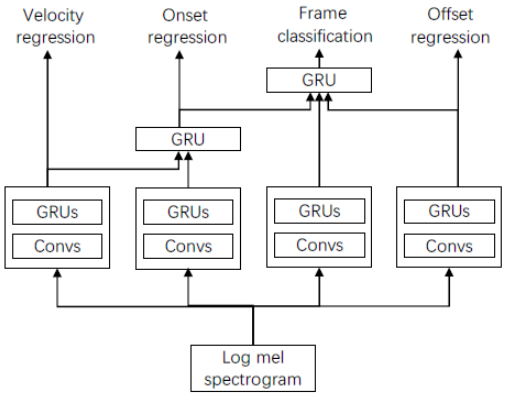

Fig xx: ByteDance Transcription Model Structure[Ref]

3.2.2 Model Performance

The ByteDance model was trained for 200k iterations using the MAESTRO dataset. Table[] 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.

Pitch-Detection Model	Frame			Note		
	P%	R%	F1%	P%	R%	F1%
ByteDance Model	88.71	90.73	89.62	98.17	95.35	96.72
ByteDance Model (Noise Labels)	84.65	91.36	87.79	98.65	94.30	96.39
Adversarial Onsets and Frames Model	93.10	89.80	91.40	98.10	93.20	95.60
Music Transformer Trained on MAESTRO	93.10	85.76	89.19	97.42	92.37	94.80

Pitch-Detection Model	Note with offset		
	P%	R%	F1%
ByteDance Model	83.68	81.32	82.47
ByteDance Model (Noise Labels)	80.59	77.09	78.77
Adversarial Onsets and Frames Model	83.50	79.30	81.30
Music Transformer Trained on MAESTRO	81.84	77.66	79.67

Table XX: Transcription Evaluation Results[Ref]
<format to be adjusted;can be removed for more important content>

We have also performed a series of experiments on the Bytedance model to achieve a deeper understanding of the model algorithm as well as its performance.

We have evaluated the ByteDance model's performance 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 [], the model takes an average of X seconds to complete the transcription of each audio.

We have used 20 piano recordings of Chopin's Nocturne to further test the model's execution speed when handling recordings with higher complexity(polyphonic recordings).

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 XX: ByteDance Efficiency

3.3 Correct/ Incorrect Note Classification
<header can be refined>

3.3.1 Combination of DTW and Sequential Mapping

For our first evaluation metric, 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 our 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.

3.3.2 Sequential Mapping

Due to the limitations of DTW (discussed in section xx), a second 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 hyperparameter 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 our 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 X shows an overview of the reference dataset.

Reference	Title	Total Number 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 X: Overview of Reference Dataset

4.1.2 Query Dataset

There are five query test sets for each reference score, as shown in Table XX. 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.

Each test set of the respective reference score is annotated accordingly to ensure proper evaluation, and the playings of a professional pianist on the annotated pieces were recorded using a digital piano and saved in M4A format.

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 XX: Overview of Query Dataset

4.2 Metrics

F1-score is used to evaluate the system performance.

4.3 Results

<further consider sub headers>

The combined algorithm of DTW alignment, sequential matching, with reset mechanism was tested with the test set consisting of all correct notes.

~ Insert and describe results xx ~

From the results, it was found that 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 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.

Limitations

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.

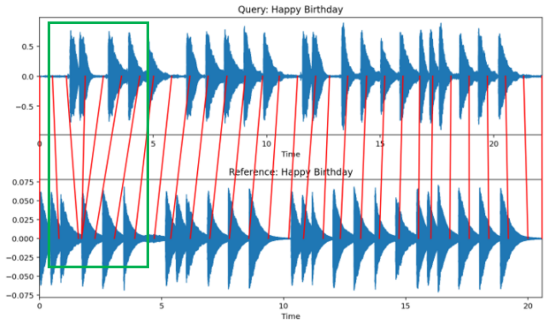

Fig xx: Many-to-one mapping exhibited by DTW due to missing notes

Another issue is fundamental to the DTW algorithm itself. As part of the formulation of DTW, one constraint imposed to make it solvable 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, the effects of which have been discussed above.

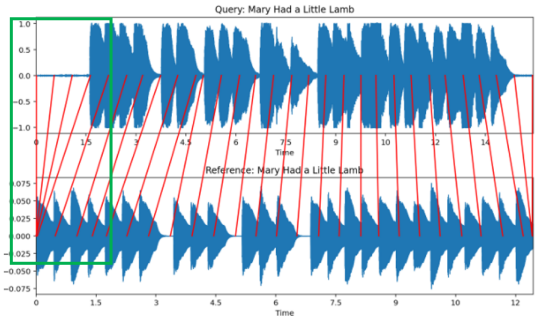

Fig xx: Many-to-one mapping exhibited by DTW due to silence at start and end of recording

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 our algorithm was tested. This algorithm uses only sequential matching, without the aforementioned DTW aspects.

<Sequential Matching subheader>

~ Insert result table for test set 1, 2, 3 ~
The sequential matching algorithm has generally good performance. However, one particular noteworthy result is that of reference 6. The accuracies for test 1 and 3 (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.

5. WEB APPLICATION

<further consider sub headers>

6. FUTURE WORK

We have identified a few areas for future optimizations if given more time and resources:

Firstly, 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 we are currently implementing in PianoES, identification of misaligned notes would not be supported. Also, our existing 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, we could work on the note detection and alignment algorithms in order to cover a wider range of user scenarios.

Furthermore, we have noticed phantom notes 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 playings would be essential to maintain a stable system performance, and this would be an important feature that we can work on in the future.

Lastly, different pianos would produce sounds that are different in colors and tones. Also, pianos might be tuned with different frequencies, making it challenging for our system to provide accurate evaluations of users' playing. This would be a possible field that PianoES can be improved in the future.

7. CONCLUSION

8. REFERENCE

<some references below do not meet the reference format that prof wanted, need to modify>

[1] S. Sofia, F. Heidi, O. Jordi, and S. Maria, "Effects of music learning and piano practice on cognitive function, mood and quality of life in older adults," in Front. Psychol, 2013.

[2] ABRSM, "Teaching, learning and playing in the uk," 2014.

[3] B. Taylor, A. Jacob, and K. Dan, "Unsupervised transcription of piano music," in Advances in Neural Information Processing Systems, 2014.

[4] L. Lele, M. Veronica, and B. Emmanouil, "Joint multi pitch detection and score transcription for polyphonic piano music," 2021.

[5] Z. Weiwei, C. Zhe, and Y. Fuliang, "Multi-pitch estimation of polyphonic music based on pseudo two dimensional spectrum," in TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, 2020.

[6] W. Siying, E. Sebastian, and D. Simon, "Identifying missing and extra notes in piano recordings using score-informed dictionary learning," in TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, 2017.

[7] T. Yoni, "A look under-the-hood of simply piano (part 2)," in https://medium.com/hellosimply/a-look-under-the-hood-of-simply-piano-part-2-3ba3cafa1bbf, 2021.

[8] W. Rom, "Musical note recognition algorithms need tuning too," in https://medium.com/hellosimply/musicalnote-recognition-algorithms-need-tuning-too-7693402edd4e, 2018.

[9] B. Emmanouil, K. Anssi, and D. Simon, "Scoreinformed transcription for automatic piano tutoring," in Networked Media and Search Systems, 2014.

[10] E. Sebastian, W. Siying, M. Meinard, and S. Mark, "Score-informed identification of missing and extra notes in piano recordings," in 17th International Society for Music Information Retrieval Conference, 2016.

Model DTW - Nic (I think this part can remove)

Certain evaluation metrics, such as note accuracy, require a basis of comparison between the player's recording against a reference. For the comparison to be fair and robust, a unified time domain would have to be created; such a mechanism would account for variability in the tempo of different players. Accounting for tempo is important as it enables the model to achieve consistent results for players of differing skill level, who might opt to play at different tempos. In such a case, metrics such as accuracy should give consistent results.

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:

$$E = \sum_{(i,j) \in M} d(u_i, v_j)$$

Where d represents some distance metric greater than 0, u_i represents a point in query sequence m, and v_j 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. This constraint, while required to solve the optimisation problem, led to several problems downstream, which will be discussed in Section xx.

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.

An algorithm widely used to solve the temporal registration problem is Dynamic Time Warping (DTW). DTW solves the problem using a path based approach, finding a path P = {p1, p2, … pL} minimizing the total distance $\sum_{(i,j)\in M} d(u_i, v_j)$. The path is found using a dynamic programming approach, whereby i and j are iterated over, and the minimum distance path is found and memoised. The optimal path can be stored as a function mapping the query time domain to the reference time domain.

Limitations (This part should keep)

The first limitation of 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 (https://core.ac.uk/download/pdf/78275924.pdf).

To overcome this limitation, our 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, we believe that our current tuning is sufficient for most scenarios.

Evaluation Metric 1: Combination of DTW and Sequential Matching

Design and Rationale

For our first evaluation metric, 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 our 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.

Upon implementation of this algorithm, it was discovered that it could not handle wrong notes being played. Once a note in a song was classified as wrong, all the remaining notes in the song were also classified as wrong. Further analysis of the implementation revealed the source of the error: when a note is not within the temporal threshold to be classified as correct, it implies that all the following notes would also fall outside the threshold even when played correctly, due to the offset resulting from the error. As such, an additional 'reset' mechanism was introduced, removing this error offset from all notes following an erroneous note.

Results

The combined algorithm of DTW alignment, sequential matching, with reset mechanism was tested with the test set consisting of all correct notes.

~ Insert and describe results ~

From the results, it was found that 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 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.

Limitations

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.l

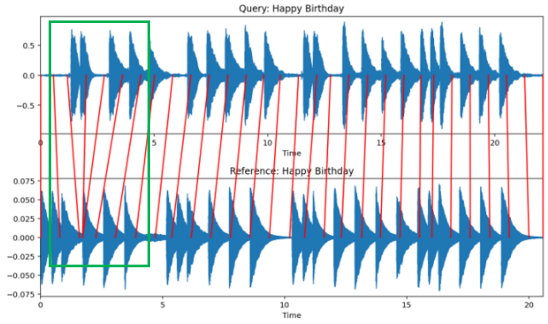

Fig xx: Many-to-one mapping exhibited by DTW due to missing notes

Another issue is fundamental to the DTW algorithm itself. As part of the formulation of DTW, one constraint imposed to make it solvable 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, the effects of which have been discussed above.

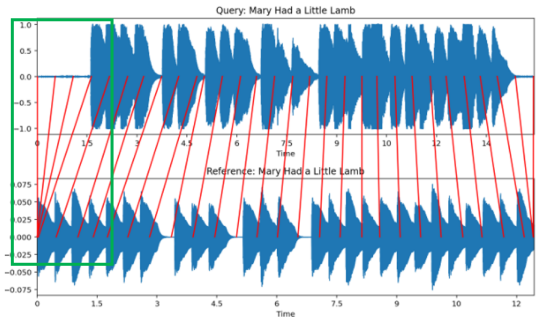

Fig xx: Many-to-one mapping exhibited by DTW due to silence at start and end of recording

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 our algorithm was tested. This algorithm uses only sequential matching, without the aforementioned DTW aspects.

Evaluation Metric 2: Sequential Matching

3. Results and Evaluation

3.1 ByteDance Transcription Model Performance

3.2 Alignment Performance

3.2.1 Combination of DTW and Sequential Matching

3.2.2 Sequential Matching

3.3 Evaluation Metrics

4. System Architecture and Design

4.1 Existing Piano Transcription System

Ewert et al. (2016)

The proposed system performs audio-score alignment by combining chroma with onsite indicator features. Data obtained after this alignment would

4.2 PianoES System Architecture

4.2.1 Front-End Design and Implementation

4.2.2 Back-End Design and Implementation

4.3 Future Optimizations

We have identified a few areas for future optimizations if given more time and resources:

Firstly, 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 we are currently implementing in PianoES, identification of misaligned notes would not be supported. Also, our existing 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, we could work on the note detection and alignment algorithms in order to cover a wider range of user scenarios.

Furthermore, we have noticed phantom notes 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 playings would be essential to maintain a stable system performance, and this would be an important feature that we can work on in the future.

Lastly, different pianos would produce sounds that are different in colors and tones. Also, pianos might be tuned with different frequencies, making it challenging for our system to provide accurate evaluations of users' playing. This would be a possible filed that PianoES can be improved in the future.

5. Conclusion

Introduction

Report

Model

Reference

Model DTW - Nic

Model BD - WT

Jon - evaluation colouring
Application - Jon + HL

Application

1. Introduction
1.1 Motivation
1.2 Problem Statement

3. Alignment Algorithm
3.1 DTW Concept+Literature Review
3.2 DTW Design in PianoES
3.3 Implementation + Issues + Solution

2. AMT
2.1 AMT Concept +Literature Review(existing approaches etc)
2.2 BD model introduction +performance evaluation
2.3 Implementation + Issues + Solutions

4. Score Evaluation Metrics
4.1 Literature Review(commonly used metrics)
4.2 PianoES metrics and rationale

5. System Design
5.1 Model Pipeline
5.2 BE architecture
5.3 FE design

6. Experiments and Results (Overall System Performance)
6.1 Experiments
6.2 Result evaluation

7. Future Improvements

8. Conclusion

1. Intro
    a. Motivation
    b. Problem statement
    c. **Related Works (literature review) -HL**
        i. Alignment
        ii. AMT - WT
        iii. Score evaluation metrics
2. Method
    a. Dataset (reference scores and queries)
    b. Data processing
    c. Modeling
        i. DTW (Rationale) - Nic
        ii. Bytedance Model - WT (BD model)
        iii. Evaluation algo - Nic
    d. Expected performance (>90%)
3. Results (Experiments +blockers + overcome challenges + solutions +final overall results)
    a. DTW (Rationale) - Nic (parameters, timing) - muspy -> 1800ms HL
    b. Bytedance Model - HL
    c. Evaluation algo - Nic, HL
    d. ….
4. System architecture and design (FE + BE)
    a. Existing piano applications - HCI
    b. Architecture (Jon, HL)
    c. FE - UI, logics (HL-UI, Jon-Logic)
    d. BE - logics
    e. Optimisation (if got time)
5. Conclusion

ABSTRACT

This paper proposed the development of a piano evaluation system to provide feedback on learner's piano performance. Given a reference piano score, the learners will record their piano playing in audio, and the system will evaluate their playing against the reference score to provide a scoring on the notes that are played correctly, missing, or extra.

1. INTRODUCTION

Learning a musical instrument is beneficial for all ages, as musical training can help strengthen memory and reading abilities, enhance concentration and develop the skill of handling multiple things at once. Among all musical instruments, piano is one of the most popular ones due to its beautiful sound and beginner-friendly nature. However, expensive piano lessons may deter learners. Also, while piano learners are doing self-practicing, the lack of real-time feedback makes their practices less effective.

1.1 Motivation

Therefore, we are motivated to develop an application which can assist piano learning by implementing automatic music transcription concepts. With this application, we aim to lower the cost of learning the piano and increase the flexibility of piano learning by providing an effective and standardized feedback system.

1.2 Problem Statement

We have identified three main challenges of developing an effective piano evaluation system. Firstly, we need to accurately detect notes played by users from the source audio file. Secondly, we need to align the user's piano playing with the reference score to compare and obtain differences between them.

Lastly, we need to determine a series of score-evaluation metrics to provide users with a fair assessment of their piano playing, and clearly illustrate to users what mistakes they have made in their playing.

1.3 Outline of Solution

The solution will focus on building a robust transcription model and the development of a web application that can provide real-time feedback and evaluation on note correctness and rhythm.

Automatic music transcription (AMT) is the task of converting raw audio into other symbolic musical representations. With the help of AMT, piano recordings from learners can be automatically transcribed into score representations and compared with the standard reference score to provide timely and accurate evaluation across all piano learners, and alleviate the high cost of piano lessons.

Scored-I nformed Transcription System

Score-informed transcription is one existing approach that takes an audio recording clip together with the reference score as inputs, then provides feedback generated from comparing these inputs. The recorded playing is firstly aligned with the reference score using alignment algorithms, then comparison results would be generated and demonstrated in a visually user-friendly format, such as piano rolls.

2.2 Alignment Algorithm

……

2.3 Piano Transcription

Automatic music transcription (AMT) is the task of transcribing audio recordings into symbolic representations, such as piano rolls, guitar fretboard charts and Musical Instrument Digital Interface (MIDI) files. AMT serves as a bridge that connects audio based and symbolic based music understanding, therefore it is an essential topic of music information retrieval (MIR). AMT systems have a wide range of possible implementations, including music production and symbolic based music information retrieval.

Piano transcription is one challenging task for AMT due to the high polyphonic nature of piano music. In this project, we aim to implement AMT in a piano music education system to assist piano learners.

Pitch-Detection Model	Frame			Note		
	P%	R%	F1 %	P%	R%	F1 %
ByteDance Model	88.71	90.73	89.62	98.17	95.35	96.72
ByteDance Model (Noise Labels)	84.65	91.36	87.79	98.65	94.30	96.39
Adversarial Onsets and Frames Model	93.10	89.80	91.40	98.10	93.20	95.60
Music Transformer Trained on MAESTRO	93.10	85.76	89.19	97.42	92.37	94.80

Pitch-Detection Model	Note with offset		
	P%	R%	F1 %
ByteDance Model	83.68	81.32	82.47
ByteDance Model (Noise Labels)	80.59	77.09	78.77

Adversarial Onsets and Frames Model	83.50	79.30	81.30
Music Transformer Trained on MAESTRO	81.84	77.66	79.67