AUTOMATIC PIANO FINGERINGS ESTIMATION USING RECURRENT NEURAL NETWORKS

Hongzhao GUAN (hguan7@gatech.edu) [1], **Zhao YAN** (zyan66@gatech.edu) [2], and **Timothy HSU** (hsut@iu.edu) [3]

[1] *H. Milton Stewart School of Industrial and Systems Engineering*, **Georgia Institute of Technology**, Atlanta, GA USA
[2] *Center for Music Technology*, **Georgia Institute of Technology**, Atlanta, GA USA
[3] *Music and Arts Technology*, **Indiana University - Purdue University Indianapolis**, Indianapolis, IN USA

ABSTRACT

Deciding piano fingerings is an essential skill for all piano players regardless of their expertise. Traditionally, pianists and piano educators first need to analyze musical scores, then they manually label the fingerings on the scores; however, this process is time-consuming and inefficient. This paper proposes a novel automatic piano fingerings estimating method by utilizing Bidirectional Long Short-term Memory (BI-LSTM) networks—a special type of Recurrent Neural Networks (RNNs). This is one of the first studies to explore the possibilities of applying deep learning to estimate piano fingerings. Together with the new method, a novel input representation is designed to capture the relations between surrounding notes. Furthermore, in addition to directly comparing the estimations with the ground-truth, this paper proposes a novel evaluation metric to assess the playability of the estimated fingerings. The results illustrate the effectiveness of the proposed method that generates playable and accurate estimated fingerings.

1. INTRODUCTION

The choice of piano fingering is a thoughtful decision process of which finger to use for playing each note of a piece, respectively. Fingering selections significantly affect the quality of piano performances, and specific fingering decisions might bring distinct musical effects [1]. For example, a sight-reading performance usually has less expressiveness than a prepared performance. Apart from the unfamiliarity with the piece, fingering selections significantly impact the performing quality [2,3]. Sight-readers often do not have the foresight to decide the best fingerings because they have not been given the chance to make thoughtful decisions [1]. The statements above only discuss fingering's effects on well-trained piano players; on the other hand, piano fingering is also essential for piano students. Beginners in piano playing need to develop excellent fingering habits; otherwise, students will not be able to adapt appropriately when they start to play difficult pieces [4]. However, it is extremely challenging for a new student to analyze a piece and generate appropriate fingerings for it; thus, it requires a significant amount of time from the piano teachers to manually label fingerings for the students. In order to save time and energy for pianists and piano teachers, a system that can automatically label accurate and playable fingerings on a piano score is desirable.

This paper proposes a novel method that automatically estimates piano fingerings using Bidirectional Long Short-term Memory (BI-LSTM) Networks [5, 6]. It considers piano fingering as sequential data and assumes there is a strong correlation between the current fingering and its surrounding fingerings. Furthermore, in order to incorporate piano fingering data with a BI-LSTM network, each note transition is converted to a 4-dimensional vector. Not only can this novel vector representation significantly reduce the amount of information from the raw fingering data, but it can also successfully preserve the correlations between the fingerings of two adjacent notes.

For the purpose of evaluation, two kinds of metrics are employed in this study. The first kind performs quantitative evaluation by comparing the estimated fingerings with the manually labeled fingerings. In contrast to quantitative evaluation, the second metric applies qualitative evaluation on estimated results. This newly designed metric assigns two scores to a sequence of fingerings and aims to evaluate how playable the sequence is. The experimental results demonstrate the novel deep learning approach can achieve high accuracy on the PIano fingerinG (PIG) dataset and generate fingering estimations with high playability.

The primary contributions of this study are threefold. First, this is the first study to focus on a deep learning approach that estimates piano fingerings. The experimental results presented in this paper demonstrate its workability. Second, by introducing the 4-dimensional vector, this study offers some novel insights into the designing of learning representation for piano fingering estimation. Finally, this study proposes a new evaluation metric that can assess the playability of a sequence of piano fingering.

2. RELATED WORK

Piano fingering has always been an essential topic in the field of piano performance and piano pedagogy. For many years, piano pedagogues attempted to study piano fingering by providing a comprehensive analysis of piano fingering rules and techniques [4, 7–9].

Computational methods to estimate piano fingering have been proposed as early as the late 1990s. Parncutt et al.

proposed one of the earliest approaches by formulating and solving a constraint-based problem [10]. In a follow-up study, Jacobs refined the previous approach by adding new constraints and improving the pitch representation [11]. Furthermore, with an online extension, Lin et al. developed an application that assists piano performances [12]. A significant extension on constraint-based methods was presented by Balliauw et al. [13]. Their method can efficiently find the optimal fingering and effectively handle polyphonic passages. Another early method for fingering estimation on monophonic passages employed Dynamic Programming (DP) [14]. The usefulness of using the number of intervals between notes has also been demonstrated in [14]. With a more generalized DP-based approach, Kasimi et al. carried out an investigation into piano fingering estimation for polyphonic passages [15]. Although the authors presented promising results in their publications, due to the absence of a well-designed metric, both approaches are only evaluated on a few musical passages.

Previous studies have also explored the possibilities of using statistical modeling on piano fingerings estimation. Yonebayshi et al. first demonstrated that piano fingering estimation for monophonic passages can be modeled by HMMs [16]. An extension of the HMM-based method is discussed in [17], this refined HMM method extends its fingering estimation abilities to both hands and polyphonic passages. The two essential parts of an HMM, i.e., transition probabilities and output probabilities, represent the moving tendency between fingers and that between hand shapes respectively. More recent attention has focused on the provision of using high-order HMMs to estimate piano fingering [18]. The high-order HMMs improved the performance by looking further into previous fingerings where a simple 1st order HMM only takes the most local fingering as a constraint. Nakamura et al. conducted experiments to demonstrate the superiority of high-order HMMs and reported a state-of-the-art performance.

Over the past decade, most research in Machine Learning tasks has emphasized the use of Deep Neural Networks (DNNs) [19]. Nakamura et al. proposed the first two DNN-based approaches for piano fingering estimation. Contrary to other machine learning tasks that are dominated by the DNNs, these two approaches are outperformed by the high-orders HMMs mentioned above [18]. However, since their study mainly focused on methods based on statistical modeling, the viability of applying deep learning to piano fingering estimation should be further explored.

In order to evaluate the proposed methods, multiple publications on this topic first presented a few examples with fingering estimation then conducted the evaluation on these results [10, 13–16]. Although these evaluations provided plenty of insights, they are not sufficient. Therefore, it is necessary to have a metric that can efficiently evaluate massive results. Nakamura et al. designed three quantitative evaluation metrics which derive from the direct comparison with the ground-truth [18]. These three metrics are adopted in this study and are introduced in Sect. 4.3.1.

3. METHODOLOGY

3.1 Baseline and state-of-the-art methods with HMM

Hidden Markov Model (HMM) is a widely-used statistical model [20]. The baseline system used in this study is a 1st order HMM proposed in [16] and introduced in Sect. 2. Other HMMs used in this study are 2nd and 3rd order HMM. As of 2021, this is the state-of-the-art approach and more details about it are summarized above in Sect. 2.

3.2 New Approach with BI-LSTM Networks

3.2.1 Input Representation

When piano players are making a fingering decision for an upcoming note, they usually need to consider at least four elements: (1) the fingering for the starting note, (2) whether the starting note is a black key on the piano, (3) whether the upcoming note is a black key, and (4) the *note-distance* between the previous note and the upcoming note. In this paper, the term *note-distance* indicates the number of semitones between two successive notes.

To represent these elements in a numerical format, 1, 2, 3, 4, and 5 are used to label a hand's fingers, from the thumb to the little finger respectively. Binary indicators are employed to denote white keys and black keys with 0 and 1 respectively. The interval between notes can be calculated by subtracting the corresponding MIDI values of the notes. Furthermore, since the fingering for octaves also fits for intervals greater than an octave (usually a jump), a *note-distance* that is greater than 12 or less than -12 will be treated as a 12 or a -12. In summary, a note transition can be represented as a 4-dimensional vector that consists of the four elements mentioned above. An example of this input presentation is shown in Fig. 1.

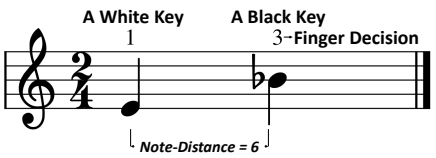

Figure 1. An example of converting a note transition to a 4-dimensional vector. The transition starts at E_4 (MIDI note No. = 64) and ends at $B^b{}_4$ (MIDI note No. = 70). The finger used to play the staring note is the thumb. Therefore, the corresponding 4-dimensional vector for this transition is $[1; 0; 1; 6]$ whose elements represent thumb, a white key, a black key, and note-distance = 6 (70 - 64), respectively.

A major advantage of this input representation is that it preserves the correlations between the fingerings of two adjacent notes and it also extracts useful information from the raw data. The second advantage of using this 4-dimensional vector is that the latter three elements of the vector guarantee the learning process is independent of musical keys and accidentals. This can be illustrated briefly by an example in Fig. 2. While the four transitions have the same *note-distance*, not all of them are playable with the same fingerings such us 2-1 and 3-1. If

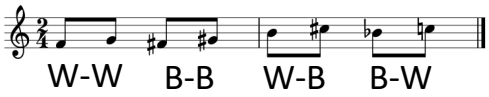

Figure 2. Each transition is represented by a pair of eighth notes. B stands for Black Keys and W stands for White Keys. All of these transitions have note-distance = 2.

transitions can only be identified by *note-distances*, like the example shown in Fig. 2, the Neural Networks will not be able to distinguish the differences between "W-W", "B-B", "B-W", and "W-B", resulting in an incapability of learning fingering patterns from music written in different keys or music with many accidentals. Therefore, prior to the training session, it is necessary to enable the Neural Networks to identify transitions that are playable with the same set of fingerings. In order to do so, for transitions with the same *note-distance*, the two binary indicators (the 2nd and the 3rd elements in the vector) are applied to further differentiate them to four categories. Thus, by applying this novel input representation, the learning process is key-independent.

3.2.2 *Modeling with Bidirectional LSTM Networks*

A fingering decision is highly dependent on the fingerings of multiple previous notes. On the other hand, it is also common for a pianist to use foresight, i.e., further analyzing a few future notes before deciding the fingering for the upcoming note [7, 8]. Therefore, to emulate the behaviors of a pianist, the estimation of the fingering for the $(n+1)$th notes can be described by the following equation:

$$f_{n+1} = \underset{i \in \{1,2,3,4,5\}}{\arg\max}\ \mathbb{P}[f_{n+1} = i | v_1, ..., v_n, v_{n+1}, ..., v_{n+m}]$$

where i stands for the fingering decision and each v represents a note-transition with the vector representation introduced in Sect. 3.2.1. It is worth pointing out that the vectors with indices greater than n have the first element set to 0, because the fingerings for their corresponding notes should remain unknown before the nth fingering is estimated. An example with $n = 2$ and $m = 2$ is provided in Fig. 3 to further illustrate this idea.

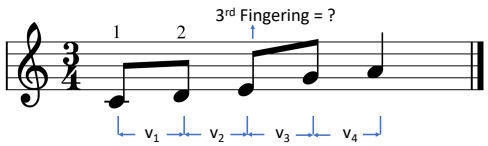

Figure 3. An example with $n = 2$ and $m = 2$. To estimate the fingering for the 3rd note, 2 previous and 2 future notes are considered. The first elements of v_3 and v_4 are 0 since they remain unknown before the 3rd fingering is estimated.

A Long Short-term Memory (LSTM) Network is an RNN that contains multiple LSTM cells and is capable of modeling sequential data [21]. Considering that notes are correlated to each other to a certain degree, an extended version of the standard LSTM network, i.e., Bidirectional

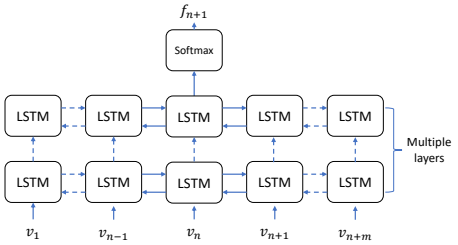

Figure 4. Diagram of an unrolled, multi-layer, many-to-one BI-LSTM network

LSTM (BI-LSTM) Network, is used in this study. As shown in Fig. 4, the BI-LSTM Network discussed in this paper has a multi-layer, many-to-one structure, and each LSTM cell contains 128 units. Furthermore, n and m are left as hyper-parameters which will be explored during the training sessions. The output of the nth time step of the unrolled BI-LSTM Network is fed into a Softmax layer to compute the probabilities of using each finger to play the $(n + 1)$th note. The maximum value of Softmax's outputs is chosen and its corresponding fingering is the desired output. The loss function employed in this model is Categorical Cross-Entropy.

There are indeed many complex Neural Network architectures that are more powerful on sequential data; however, given the small size of the input representation and the limited amount of publicly available training data in this field, a simpler architecture is more appropriate for this study.

3.2.3 *Handling Ornaments and Chords*

There are multiple kinds of ornaments in piano playing and they are handled by the proposed approach. The network treats grace notes, such as the appoggiatura and the acciaccatura, as regular notes. For trills, the network only considers one pair of alternation because a trill is constructed by multiple alternations between the indicated note and the note above. Mordents and turns are first converted to regular notes then passed to the network. Additionally, glissandos are ignored.

When a chord appears in the middle of single notes, the BI-LSTM Network cannot directly process it. To carry out an uninterrupted learning process, only the lowest note from the chord is preserved; thus, all chords are transferred to single notes beforehand. After obtaining the estimations from the BI-LSTM Network, the fingerings for the chord will be generated separately and replace the previous es-

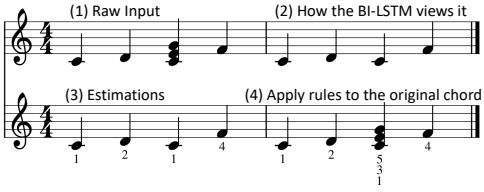

Figure 5. An example when a chord appears in the middle of notes. The estimation process is divided into 4 steps.

timation for the lowest note. Although this is not an optimal solution under many circumstances, it is valid since the fingerings for a chord can be easily deducted by a limited number of pre-defined fingering rules [7]. It is worth mentioning that a chord is not simply treated as an arpeggio because a chord does not allow finger crosses. Fig. 5 presents an example to illustrate a scenario when a chord is involved.

4. EXPERIMENTAL SETUP

4.1 Dataset

The PIG dataset was developed by Nakamura et al. [18] and is available online. [1] This dataset contains 150 pieces of piano music with labeled fingerings. Most of these pieces are well-known to the audiences and were composed by 24 notable composers such as J. S. Bach, Mozart, and Chopin. This dataset is separated into four subsets. Each of the first three subsets consists of 10 pieces composed by J. S. Bach, Mozart, and Chopin, respectively. The developers named the 4th subset as the "miscellaneous subset," and it includes 120 different pieces composed by 24 composers. It should be observed that there are multiple pieces in subset 4 that are composed by the three composers mentioned above, but no pieces overlap subset 1, 2, 3, or 4.

Since multiple pianists are asked to label the scores, 69 out of the 150 pieces have more than one fingering version, resulting in a total of 309 pieces with different fingerings. All 30 pieces in the first 3 subsets have multiple fingering versions, totaling in 40, 60, and 50 pieces with different fingerings in subsets 1, 2, and 3 respectively. Among the 120 pieces in the miscellaneous subset, only 39 of them have 2 versions; thus, in subset 4, there are 159 pieces with different fingerings in total.

4.2 Experiments

In order to investigate the BI-LSTM Network's performance on this task, three experiments are carried out on the PIG dataset. For each of the experiments, the BI-LSTM Network is compared with a baseline HMM (1st order), a 2nd order HMM, and the existing state-of-the-art—a 3rd order HMM. Table 1 compares the differences between the three experiments. Given that the pieces in the dataset contain considerably more chords in their left-hand parts, this study only considers the right hand notes for the experiments. However, the same method can be applied to the left-hand fingerings if more left-hand data are available.

These experiments are designed to achieve two following research goals: (1) to demonstrate the practicability of using BI-LSTM Network on piano fingerings estimation using the results from all experiments and (2) to investigate the BI-LSTM Network's generalization ability on pieces that are written by representative composers from three distinct periods—Baroque (subset 1, Bach), Classical (subset 2, Mozart), and Romantic (subset 3, Chopin).

[1] PIG Dataset: `http://beam.kisarazu.ac.jp/~saito/research/PianoFingeringDataset` Last Access Date: 2021-08-21

The number of LSTM layers in the network is fixed at 3. The Adam Optimizer is used during the training session. The hyper-parameters, i.e., n and m explained in Sect. 3.2, are tuned with a validation set which is randomly separated from the training set. Prior to isolating the validation set, the pieces of the training set are pre-processed to segments that contain $m + n + 1$ notes, then 15% of these segments are randomly selected as the validation set.

Furthermore, the 1st, 2nd, and 3rd order HMM methods are directly adopted from [18] using the corresponding source code. [2] The HMMs are trained and tested with the same experimental setup introduced in this section.

	Training Set		Test Set	
Exp.	subsets	# RH Notes	subsets	# RH Notes
1	2,3,4	34,660	1	8,085
2	1,3,4	35,437	2	7,308
3	1,2,4	36,073	3	6,521

Table 1. The experimental setups for the three experiments. RH stands for right hand.

4.3 Metrics

4.3.1 Quantitative Metrics

Since the decisions to piano fingerings are not unique, a metric that can evaluate the estimation with multiple ground-truths is desirable. Three previously proposed metrics, M_{gen}, M_{high}, and M_{soft}, are designed to achieve this goal and are adopted in this study [18]. M_{gen} computes the averaged match rate with multiple ground-truths and M_{high} returns the match rate with the closest ground-truth. The 3rd metric, M_{soft}, considers an estimated fingering as a correct estimation if it matches with any ground-truth. After computing the three accuracy values for each piece, three averaged values over pieces are reported.

4.3.2 Novel Qualitative Metric with Fingering Rules

note-dist.	W/B	fingering list	note-dist.	W/B	fingering list
3	W-W	1-2, 1-3, 1-4, 1-5, 2-1 2-3, 2-4, 2-5, 3-4, 3-5, 4-5	-3	W-W	5-4, 5-3, 5-2, 4-3, 4-2, 4-1 3-1, 3-2, 2-1, 1-2, 1-3
	W-B	1-2, 1-3, 1-4, 1-5, 2-3 2-4, 2-5, 3-4, 3-5, 4,5		W-B	5-4, 5-3, 5-2, 4-3, 4-2, 4-1 3-2, 3-1, 2-1, 1-2, 1-3, 1-4
	B-W	1-2, 1-3, 1-4, 2-1, 2-3 2-4, 3-1, 3-4, 3-5, 4-5		B-W	5-4, 5-3, 5-2, 5-1, 4-3 4-2, 4-1, 3-2, 3-1, 2-1
	B-B	1-2, 1-3, 1-4, 1-5, 2-3 2-4, 2-5, 3-4, 3-5, 4-5		B-B	5-4, 5-3, 5-2, 5-1, 4-3 4-2, 3-2, 3-1, 2-1, 1-2, 1-3

Table 2. Proper right hand fingering for each category when *note-distance* = 3 or -3.

note-dist.	W/B	fingering list
> 0	W-W or B-W	3-2, 4-2, 4-3, 5-1, 5-2, 5-3, 5-4
	W-B or B-B	2-1, 3-2, 4-2, 4-3, 5-1, 5-2, 5-3, 5-4
< 0	W-W	1-5, 2-3, 2-4, 2-5 3-4, 3-5, 4-5
	W-B or B-B	1-5, 2-3, 2-4, 2-5 3-4, 3-5, 4-5
	B-W	1-5, 1-4, 2-3, 2-4, 2-3-4, 3-5, 4-5
0	any	N/A

Table 3. Erroneous right hand fingering for all categories. Fingerings neither proper nor erroneous are considered as "not-ideal".

[2] Statistical Learning and Estimation of Piano Fingering: `https://statpianofingering.github.io/demo.html` Last Access Date: 2021-08-21

A qualitative rule-based evaluation metric is introduced in this section. Unlike a quantitative metric which evaluates the estimation accuracies, this novel metric is designed to evaluate the playability of the estimated fingerings by assessing them with one of the three labels—proper, non-ideal but playable, and erroneous. Once the assessment on all estimations is undertaken, the proper and erroneous rates are computed. An estimated fingerings sequence with high playability is expected to gain a high proper rate and a low erroneous rate.

As discussed in Sect. 3.2.1, transitions with the same *note-distance* can be classified into four categories using two binary indicators (indicating white key or black key), and all transitions in the same category share the same set of fingerings rules. Therefore, for each category of note transitions, proper and erroneous fingerings are listed, and unlisted fingerings are considered as "non-ideal but playable". Table 2 presents proper fingerings for each of the four categories when *note-distance* = 3 or -3. Table 3 summarizes erroneous fingerings under all circumstances. Some fingerings, although assessed as erroneous by the rules, are still valid when the played note is a staccato or the first note of a new phrase. Since the PIG dataset does not indicate the starting of a musical phrase or staccatos, applying the rules on these special cases is inevitable. Furthermore, the pieces in the test sets are relatively short and only contain a few staccatos; thus, this qualitative metric is applicable to those three specific test sets. Similar rules can be designed and applied to left-hand piano fingerings, they are not explored in this study due to the reason explained in Sect. 4.2.

5. RESULTS

Experiments are conducted only on notes for the right hand. Twenty trials are carried out for all experiments and averaged values over trials are reported. Additionally, the Deep Learning approach is incapable of estimating chord fingerings; thus, chords are eliminated during evaluation.

Tables 4 and 5 present the results obtained from Experiment 1 and 2, respectively. The first observation is, the BI-LSTM Network outperforms all three HMMs on the three quantitative metrics. It can thus be suggested that the network successfully learned the fingerings patterns from the information preserved by the input representations. Secondly, by observing the proper rates (% P), it is plausible that the fingerings generated by the BI-LSTM have lower playability. Nevertheless, because BI-LSTM also gained low erroneous rates (% E), it can be implied that there are approximately 10% of the generated fingerings that are "non-ideal but still playable".

Fig. 6 illustrates an example estimated by the BI-LSTM Network. Beside a sequence of playable fingerings, what stands out in the example is, the model learns from the training data that the thumb should be usually avoided when playing a black key; thus, the models first assigns the 2nd finger on the $G^{\#}$ then suggests a thumb-under on the next note. The fingerings shown in the second measure are somewhat counter-intuitive, because they do not follow the traditional broken triad fingering. Because the next note of the piece has an even higher pitch, the network successfully looks ahead to the next notes and makes proper decisions for the upcoming one. This finding suggests that the architecture of the BI-LSTM Network has positive effects on the learning process.

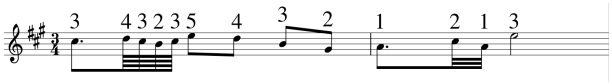

Figure 6. Estimated fingerings for a segment composed by Mozart. (Piano Sonata No. 18 K. 576, 2nd movement)

The last experiment's results are shown in Table 6. While the BI-LSTM still outperforms the baseline, no significant differences were found between the BI-LSTM approaches and the high-order HMM approaches. Another observation is, all four approaches gained relatively low proper rates but reached high accuracies. A possible explanation for this might be that the hidden fingering patterns for piano music written in different eras or by different composers might be diverse.

Method	M_{gen}	M_{high}	M_{soft}	% P	% E
1st HMM	57.2	64.0	84.4	94.1	0.3
2nd HMM	57.7	65.4	85.7	95.0	0.2
3rd HMM	59.1	66.2	86.4	94.6	0.3
BI-LSTM	60.5	68.3	86.9	88.7	0.1

Table 4. Results of Experiment 1: Test set is the Bach subset (subset 1). P and E stand for proper and erroneous

Method	M_{gen}	M_{high}	M_{soft}	% P	% E
1st HMM	57.2	65.3	85.6	92.9	0.8
2nd HMM	58.4	66.8	86.5	92.8	0.7
3rd HMM	58.7	63.3	82.6	93.5	0.5
BI-LSTM	62.1	67.2	87.2	89.0	0.6

Table 5. Results of Experiment 2: Test set is the Mozart subset (subset 2). P and E stand for proper and erroneous

Comparing the results of the three experiments, it can be seen that the Deep Learning model shows different generalization abilities on piano music composed by different composers. Overall, these results indicate that the BI-LSTM Network is capable of producing accurate estimation with high playability. On the other hand, there was no evidence that the BI-LSTM Network has a positive influence on playability. These observations may support the hypothesis that there exists a trade-off between matching the ground-truth and generating playable fingerings. The next observation from all three experiments is that the optimal n values are greater than the best m values after hyper-parameter tuning. The optimal (n, m) for the three experiments are: $(2, 8)$, $(2, 10)$, and $(2, 10)$. These findings suggest that future notes have stronger effects on the decisions of the upcoming fingerings.

6. CONCLUSION

This paper first proposed a novel approach using Deep Learning to estimate piano fingerings. The experimental

Method	M_{gen}	M_{high}	M_{soft}	% P	% E
1st HMM	63.8	69.1	81.5	80.2	0.0
2nd HMM	69.2	74.6	86.1	81.2	0.1
3rd HMM	71.0	76.3	87.8	81.3	0.0
BI-LSTM	67.9	73.4	83.6	80.5	0.1

Table 6. Results of Experiment 3: Test set is the Chopin subset (subset 3). P and E stand for proper and erroneous

results demonstrated that the proposed approach can reach high accuracies on the PIG dataset and generate fingering estimations with high playability. This paper also introduced a new input representation and a new evaluation metric. The insights gained from the representation and the metric may be of assistance to future studies on instruments fingerings estimation.

Further research on the following two topics would be a useful way of studying piano fingerings estimation. First, future studies could investigate the hidden fingerings patterns of different composers or different genres. Secondly, while the current qualitative metric effectively evaluates the estimated fingerings, the design of qualitative metrics which cover more musical aspects such as tempo and rhythm will be a significant and necessary research topic for piano fingering estimation.

Acknowledgments

We thank Eita Nakamura, Yasuyuki Saito, Kazuyoshi Yoshii, and Shinichi Furuya for developing and sharing the PIG dataset with the community. We also want to thank Eita Nakamura, Yasuyuki Saito, and Kazuyoshi Yoshii for sharing the source codes of the HMM-based approaches.

7. REFERENCES

[1] E. Clarke, R. Parncutt, M. Raekallio, and J. Sloboda, "Talking fingers: An interview study of pianists' views on fingering," *Musicae Scientiae*, vol. 1, no. 1, pp. 87–107, 1997.

[2] J. A. Sloboda, E. F. Clarke, R. Parncutt, and M. Raekallio, "Determinants of finger choice in piano sight-reading." *Journal of experimental psychology: Human perception and performance*, vol. 24, no. 1, p. 185, 1998.

[3] R. Parncutt, J. A. Sloboda, and E. F. Clarke, "Interdependence of right and left hands in sight-read, written, and rehearsed fingerings of parallel melodic piano music," *Australian Journal of Psychology*, vol. 51, no. 3, pp. 204–210, 1999.

[4] G. Kochevitsky, *The art of piano playing: a scientific approach*. Alfred Music, 1995.

[5] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] J. Musafia, *The art of fingering in piano playing*. MCA Music, 1971.

[8] H. Neuhaus, *The art of piano playing*. Kahn and Averill, 2008.

[9] P. Roskell, *The art of piano fingering: a new approach to scales and arpeggios*. LCM Publications, 1996.

[10] R. Parncutt, J. A. Sloboda, E. F. Clarke, M. Raekallio, and P. Desain, "An ergonomic model of keyboard fingering for melodic fragments," *Music Perception*, vol. 14, no. 4, pp. 341–382, 1997.

[11] J. P. Jacobs, "Refinements to the ergonomic model for keyboard fingering of parncutt, sloboda, clarke, raekallio, and desain," *Music Perception*, vol. 18, no. 4, pp. 505–511, 2001.

[12] C.-C. Lin and D. S.-M. Liu, "An intelligent virtual piano tutor," in *Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications*, 2006, pp. 353–356.

[13] M. Balliauw, D. Herremans, D. Palhazi Cuervo, and K. Sörensen, "A variable neighborhood search algorithm to generate piano fingerings for polyphonic sheet music," *International Transactions in Operational Research*, vol. 24, no. 3, pp. 509–535, 2017.

[14] M. Hart, R. Bosch, and E. Tsai, "Finding optimal piano fingerings," *The UMAP Journal*, vol. 21, no. 2, pp. 167–177, 2000.

[15] A. Kasimi, E. Nichols, and C. Raphael, "A simple algorithm for automatic generation of polyphonic piano fingerings," in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2007, pp. 355–356.

[16] Y. Yonebayashi, H. Kameoka, and S. Sagayama, "Automatic decision of piano fingering based on a hidden markov models." in *IJCAI*, vol. 7, 2007, pp. 2915–2921.

[17] E. Nakamura, N. Ono, and S. Sagayama, "Merged-output hmm for piano fingering of both hands." in *ISMIR*, 2014, pp. 531–536.

[18] E. Nakamura, Y. Saito, and K. Yoshii, "Statistical learning and estimation of piano fingering," *Information Sciences*, vol. 517, pp. 68–85, 2020.

[19] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[20] L. Rabiner and B. Juang, "An introduction to hidden markov models," *ieee assp magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[21] J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. of the national academy of sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.