A TIMBRE-BASED APPROACH TO ESTIMATE KEY VELOCITY FROM POLYPHONIC PIANO RECORDINGS

Dasaem Jeong, Taegyun Kwon, Juhan Nam
Graduate School of Culture Technology, KAIST, Korea
{jdasam, ilcobo2, juhannam} @kaist.ac.kr

ABSTRACT

Estimating the key velocity of each note from polyphonic piano music is a highly challenging task. Previous work addressed the problem by estimating note intensity using a polyphonic note model. However, they are limited because the note intensity is vulnerable to various factors in a recording environment. In this paper, we propose a novel method to estimate the key velocity focusing on timbre change which is another cue associated with the key velocity. To this end, we separate individual notes of polyphonic piano music using non-negative matrix factorization (NMF) and feed them into a neural network that is trained to discriminate the timbre change according to the key velocity. Combining the note intensity from the separated notes with the statistics of the neural network prediction, the proposed method estimates the key velocity in the dimension of MIDI note velocity. The evaluation on Saarland Music Data and the MAPS dataset shows promising results in terms of robustness to changes in the recording environment.

1. INTRODUCTION

Polyphonic piano transcription is one of the most active research topics in automatic music transcription [1]. However, the absolute majority of piano transcription algorithms so far have been concerned with detecting the presence of notes in term of pitch (or note number), onset and duration, while ignoring note dynamics, which is expressed by key velocity on piano.

Along with tempo, dynamics is a key feature that produces a musical "motion" [19]. Previous studies on piano performance analysis employed dynamics as one of two main features of performance characteristics in [22, 25]. Another study showed that, if dynamics is estimated for individual notes, a finer analysis is achievable [21].

There have been a few works that challenged the task of estimating individual note dynamics. To best of our knowledge, the first attempt was made by Ewert and Müller who tackled the problem using a parametric model of polyphonic piano notes [7]. Our previous work estimated the note intensity using score-informed non-negative matrix factorization (NMF) in various training strategies [15]. Szeto and Wong used a sinusoidal model to separate chords tones into individual piano tones and estimated the note intensity as part of the source separation task [23].

All of them basically estimate individual note dynamics according to energy magnitude or loudness of the notes. However, this approach has an essential limitation in that a note produced by a certain key velocity can be recorded in different sound levels depending on the recording conditions. For example, a *pianissimo* note can be recorded loudly or a *forte* note can be quietly, depending on the input gain of the recording device or the distance from the microphone.

In this paper, we challenged to overcome this limitation by focusing on differences in timbral characteristics caused by the key velocity. According to previous research, loudness and tone of a piano note are uniquely determined by the velocity of the hammer at the time it strikes the strings [12]. This implies that the key velocity can be inferred not only from the loudness but also from the timbre of the note, assuming that the hammer velocity can be approximated by the key velocity. This idea was explored in [14] where a piano note shows different timbral characteristics such as a spectral envelope or inharmonicity, depending on the key velocity. While the previous work focused on single notes, we study it for polyphonic music.

The proposed system consists of three parts: an NMF module for note separation and intensity estimation, a neural network to discriminate key velocity, and intensity-to-velocity calibration using the results from the two modules. The NMF module is based on score-informed settings from [15] and [24]. After the decomposition of the audio spectrogram, we reproduce the note-separated spectrogram from the NMF module. The neural network takes the note-separated spectrogram as input and estimates its key velocity. The third part obtains proper mapping parameters between note intensity and key velocity using the distribution of velocity estimation from the neural network, and finally estimate individual key velocity in the dimension of MIDI note velocity. We evaluate the proposed method on Saarland Music Data and the MAPS dataset and show promising results in terms of robustness to changes in the recording environment.

The rest of paper is structured as follows. In Section 2 we introduce the scope of our work and define the terms

that represent dynamics of a piano note. Section 3 summarizes the related works. In Section 4 we explain the NMF and neural network framework. The experiment and result are explained in Section 5 and 6. Finally, the conclusion is presented in Section 7.

2. BACKGROUND

To provide better understanding of the task and scope in this research, we first review key terms and define the problem that we attempt to solve.

2.1 Term Definitions

Note intensity is the term that represents the magnitude of acoustical energy of a note. It can be defined as sound-pressure level (SPL) [10] or the sum of spectral energy as in [7, 15]. Since the intensity is an acoustical feature, it is highly variable by the recording condition. For example, note intensity can be changed by simple post-processing such as gain adjustment. Therefore, the intensity of each note is comparable only when the recording conditions are consistent.

Key velocity refers to the kinetic velocity of the piano key and it is closely connected to the hammer velocity. It can be measured by detecting the elapsed time when the hammer shank passes two fixed points [10]. Unlike the note intensity, the key velocity is a feature measured directly from the mechanical movement, hence independent from the acoustic recording environment. If the recording condition is constant and the sympathetic resonance is ignored, the mapping between key velocity and note intensity for each pitch is linear [10].

MIDI velocity is the term that represents the key velocity in the MIDI format. It is a one-byte integer value between 0 and 127 inclusive in the note messages. Computer-controlled pianos or MIDI-compatible keyboards have their own mapping of key velocity to MIDI velocity.

2.2 Problem Definition

The aim of this study lies in estimating note key velocity in terms of MIDI velocity. Although our previous work attempted to produce the result in MIDI velocity, the method requires an additional data for intensity-to-velocity calibration with the same piano and recording condition [15]. In a real-world situation, however, it is almost impossible to obtain such mapping for a target recording. Instead of employing a target-suited training set, our work aims to learn a proper intensity-to-velocity mapping directly from a target audio recording.

One of the obstacles in the task is that most datasets represent the key velocity with MIDI velocity and the mapping between the two varies depending on the piano or keyboard model. To focus on the relation between timbre and key velocity in this study, we fix the key-to-MIDI velocity mapping by employing only one piano model but different recording conditions during the evaluation. However, we evaluate the trained model on recordings with a different piano to see how it generalizes. The details will be explained in the evaluation section.

3. RELATED WORKS

Our proposed method is based on the NMF framework from [15] but expand it by employing a recent work by Wang *et al* [24]. One of the main limitations in the NMF framework is that it is difficult to model the timbre changes over time. For example, the NMF model used in [8] and [15] assumes the spectral template of each pitch does not change over time. To overcome this limitation, Wang *et al* suggested using multiple spectral templates per pitch in NMF for piano modeling. This NMF model was adopted in our proposed system and will be discussed in more detail in the next section.

Identifying key velocity by its timbre can be compared to identification of musical instruments. The earlier works used various hand-crafted audio features [6, 14]. Recently, deep neural network has become a popular solution for this task [2, 11], which takes spectrograms or mel-frequency cepstral coefficients as input. There are a few work interested in timbral difference by the velocity [4, 14] but they did not aim to distinguish these difference explicitly.

Our task can also be compared to instrument identification in polyphonic audio. One of typical solutions for this task is using source separation and then handling it as monophonic audio sources. Heittola *et al.* suggested a framework with NMF-based source separation module [13]. Similar to this work, our method also employs NMF-based source separation. But we use the neural networks instead of the Gaussian mixture model to identify the separated sources.

4. METHOD

Our proposed system consists of three parts as shown in the Figure 1. The first part is score-informed NMF that factorizes the spectrogram of audio recording into note-separated spectrogram for every note in the score. This also returns the intensity of each note. The second part is neural network (NN) that takes the note-separated spectrogram and estimates the key velocity. The third part is intensity-to-velocity calibration which is conducted by comparing the estimated velocity from the NN module and the intensity from the NMF module on their distributions.

4.1 Note Separation

The first part of our framework is based on NMF, a matrix factorization for non-negative data which is usually spectrogram in audio processing domain.

Let us denote a given spectrogram as $\mathbf{V} \in \mathbb{R}_{\geq 0}^{F \times T}$, where F is the number of frequency bins and T is the number of time frames. With NMF, the spectrogram can be factorized into multiplication of two matrices $\mathbf{W} \in \mathbb{R}_{\geq 0}^{F \times (P \cdot R)}$ and $\mathbf{H} \in \mathbb{R}_{\geq 0}^{(P \cdot R) \times T}$ where P denotes the number of pitch in semitone and R denotes the number of spectral basis per

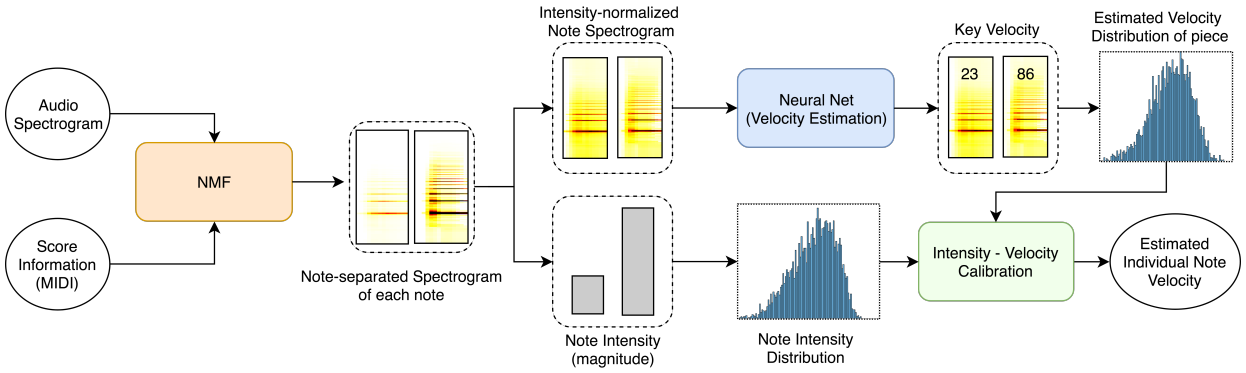

Figure 1. A diagram of the proposed system.

pitch. By doing so we can decompose the input spectrogram with spectral templates bases $\mathbf{W}$ and the activation of the bases over time $\mathbf{H}$.

To clarify the relationship between spectral basis and pitch, we will follow the similar notation presented in [24], denoting $\mathbf{W}_{f,p,r} := \mathbf{W}_{f,(p-1)\cdot R+r}$ and $\mathbf{H}_{p,r,t} := \mathbf{H}_{(p-1)\cdot R+r,t}$ as below:

$$\mathbf{V}_{ft} = \sum_{p,r} \mathbf{W}_{f,p,r}\mathbf{H}_{p,r,t} \quad (1)$$

where $f \in [1, F]$, $t \in [1, T]$, $p \in [1, P]$, and $r \in [1, R]$ are index of frequency bin, time frame, pitch, and spectral basis in a pitch, respectively.

4.1.1 NMF Modeling

We employ an NMF model that learns multiple time-frequency patterns instead of single spectral templates [24], which was applied to the score-informed AMT task. This model captures various timbre of the same pitch and temporal evolution of timbre, which is a necessary part of our task. Since the main contribution of our paper lies on the velocity estimation by combining of the NMF and NN results, the following section will mainly explain several differences in the implementation. The details are found in [24].

Considering that an NMF model can be configured mainly by the number of basis, initialization method, and additional constraints with corresponding update rules, Wang *et al.*'s model for piano recording [24] is different from the previous models used in [8,9,15] in three aspects.

First, they suggested multi-basis per pitch so that each pitch has R number of corresponding bases. The previous models represent a piano note by the combination of percussive (onset) and harmonic (sustain) basis for the whole note duration. Since there is only one harmonic basis for each pitch, the spectral shape of the note does not change over time. This assumes that the most important timbre feature is constant in the sustain part within the single note as well as for different key velocities. But the multi-basis model can handle this subtle change of timbre by using multiple bases with different activation ratios.

Second, employing the multi-basis model requires a different initialization method for matrix $\mathbf{W}$ and $\mathbf{H}$. To model temporal progression of piano timbre, the r-th basis was initialized to be active after the $(r-1)$-th basis of the same pitch. Since the pitch bases are activated sequentially, they can model temporal evolution of the note tone. As the pitch bases are differed by their activation initialization, they also have different spectral characteristics. Among R bases of a pitch, the first basis handles percussive element and the the second to the last represent harmonic elements in the temporal order. In addition, the harmonic area is set to be tapered as the rank index r increases. This makes the earlier bases include more inharmonicity.

Third, Wang *et al.*'s model suggested several additional costs for the multi-basis model. They include a soft constraint, temporal continuity, and energy decay in the template matrix. Among the suggested costs, we did not employ the decaying cost for $\mathbf{W}$, which encourages smooth decrease of energy in spectral templates in $\mathbf{W}$. We found that our system works better with L1 normalized $\mathbf{W}$ so that the magnitude feature is assigned only to $\mathbf{H}$. We followed the NMF costs and update function strictly except that we ignore the decaying cost term by assign 0 to β_3.

For better intensity estimation, we previously suggested using power spectrogram, instead of linear magnitude spectrogram [15]. We also showed that using synthesized monophonic scale tones helps to learn spectral template. Based on this observation, our system also uses power spectrogram and synthesized piano scale. Another difference with [24] is post-updating of $\mathbf{H}$. After the update converges, we set all constraints on $\mathbf{H}$ to zeros and update $\mathbf{H}$ for ten times with fixed $\mathbf{W}$ so that our final reproduction can resemble the original gain.

The NMF module reproduces note-separated spectrogram $\hat{\mathbf{V}}^{(n)}$ for each note n in the score by multiplying the spectral bases of note's pitch and its activation over note's duration. The note intensity is defined as the maximum activation of $\hat{\mathbf{V}}^{(n)}$, which can be represented as $\max(\sum_f \hat{\mathbf{V}}^{(n)}_{ft})$. Then, we reproduce $\hat{\mathbf{V}}^{(n)}$ again around the time frame of the maximum activation and store it for the input for the neural network. This helps to fix the size of NN's input and maintain the relative position of each element in the cropped spectrogram.

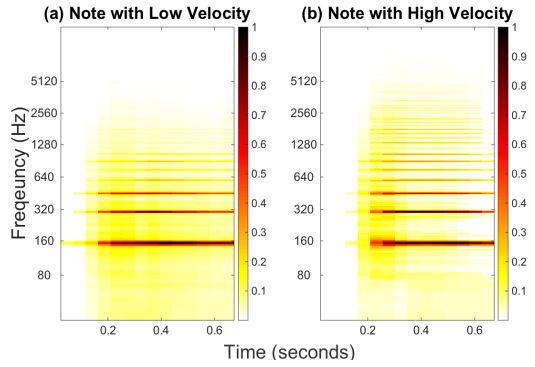

Figure 2. Comparison of the intensity-normalized note-separated spectrogram with different MIDI velocities. The spectrogram was reproduced from polyphonic piano recording (SMD). The MIDI note number is 50 and the MIDI velocities were 14 and 95, respectively.

4.2 Velocity Estimation

The neural network (NN) model takes the note-separated spectrograms from the NMF module as input and estimates the velocity of each note. The note-separated spectrogram is converted to a log-frequency spectrogram before it is used for the input of the NN module. The frequency resolution is set to 25 cent and the frequency range is from 27.5 Hz (the lowest pitch of piano) to 16.7 kHz (two octave higher than the highest pitch of piano), resulting in 445 frequency bins. After some preliminary test, we used 14 frames as input size. The spectrogram magnitude is normalized by the maximum value so that every entry in the spectrogram lies between 0 and 1 the as shown in the Figure 2.

The neural network consists of 5 fully-connected hidden layers and each layer has 256 nodes. Every hidden layer uses SELUs as an activation function [17]. Applying SELUs aims to stabilize the network from internal covariance shifting without any additional complexity.

The loss function is set to mean square error of key velocity estimation, approaching the task as a regression problem. We also attempted to use softmax as a classification problem but the result was slightly worse. We used Adam optimization [16] with initial learning rate of 1e-4, and early stopping on the validation set.

4.3 Intensity-to-Velocity Calibration

The NN module provides an absolute degree of note dynamics but the relative magnitude between each note from the NMF results is more stable than that from the NN results. Therefore, we combine the two results to find better estimation.

As described in Section 1, intensity is affected by both key velocity and recording condition. One cannot distinguish whether the high intensity from the NMF is caused by strong strike of hammer or high gain in the recording device. Therefore each recording condition needs its own mapping parameter.

Also, the intensity-velocity relation depends on a piano or a keyboard model [3]. Our previous study showed that the MIDI velocity of a note can be approximated by a linear relationship with the log value of the intensity $\mathrm{Int}(n)$, so that $\mathrm{Vel}(n) = a{\cdot}\log(\mathrm{Int}(n)){+}b$ for the Disklavier, which we use for the evaluation [15]. However, we need to know intensity-paired velocity in the target recording condition, which is not available in real-word recordings.

Our solution is estimating it from the overall velocity distribution of each piece from the NN module. If we assume the outcome velocity has a distribution with mean μ_V and standard deviation σ_V for each piece, we can obtain the mapping parameters by comparing it with the distribution of log of intensity, $\mu_{\log(I)}$ and $\sigma_{\log(I)}$. Then, the mapping parameter a and b correspond to $\sigma_V/\sigma_{\log(I)}$ and $\mu_V - (\sigma_V/\sigma_{\log(I)})\mu_{\log(I)}$, respectively, with the assumption that every note has the same mapping parameters. Note that this neglects the note-specific difference of intensity-to-velocity mapping parameter. The error caused by this assumption will be also explained in Section 6.

Our system takes the result of the NN module to estimate μ_V and σ_V for each piece. The estimation can be also done by a simple global setting. During the evaluation, we used this scheme as a baseline to compare with our NN model.

5. EXPERIMENT

5.1 Experiment I: SMD

We used Saarland Music Dataset (SMD) MIDI-Audio Piano Music [18] for the evaluation. The dataset consists of fifty pairs of audio and MIDI recordings of performance on Yamaha Disklavier DCFIIISM4PRO. The MIDI files of SMD contain every movement of piano key and pedal in high reliability, thus providing the ground truth of note dynamics in MIDI velocity.

The previous work pointed out that the recording condition of each piece in SMD is differed by its recording date [15]. Therefore, the intensity-to-velocity mapping had to be obtained separately for each subset of pieces that share the same recording condition. The difference in intensity-to-velocity mapping in SMD is represented in Figure 3. Since the goal of the proposed system is to estimate key velocity robustly against changes in the recording environment, such different recording conditions are ideal for evaluating this task.

We evaluate whether the proposed system can handle different recording conditions and estimate correct velocity distributions. We used fifteen pieces recorded in the year of 2011 as a test set, and other thirty-five pieces as a training set, which was recorded during the year of 2008 and 2010.

To evaluate the exact performance and usefulness of the NN module, we also present two upper boundary models and a baseline model. The first upper boundary assumes that the system obtained proper mapping parameters for every individual pitch from other pieces in the same test set, as in [15]. The second upper boundary assumes that our NN module guessed correct estimation of velocity dis-

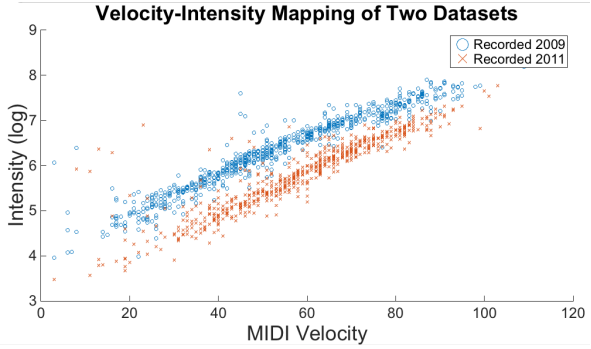

Figure 3. The difference in velocity-intensity mapping between two subsets from SMD. Each point represents a single note with MIDI note number 50. The notes recorded in 2009 show higher intensity compared to the notes recorded in 2011 given the same velocity.

tribution. In this upper boundary, we employed the ground truth of velocity distribution for each piece. The baseline is using global mean and standard deviation values. Based on the statistics of training set, we used $\mu_V = 57.87$ and $\sigma_V = 16.25$.

The evaluation measure is an absolute error of velocity between ground truth and estimated value. In MIDI velocity dimension, absolute error is a more meaningful criterion than relative error because MIDI velocity is already a logarithm of the intensity. We used the average of absolute velocity error in a piece, which can be represented as $\text{Err} = \sum_n^N |V_{\text{GT}}(n) - V_{\text{Est}}(n)| / N$, where $V_{\text{GT}}(n)$ and $V_{\text{Est}}(n)$ are ground truth velocity and estimated velocity of the n-th note in a piece, respectively.

5.2 Experiment II: MAPS

We also evaluate our NN module on unseen data to see whether the NN can learn generalized piano timbre from the training set. To this end, we designed another experiment with the MAPS database [5], which was recorded with a different piano and recording conditions.

From the MAPS dataset, we used two subsets performed by Yamaha Disklavier Mark III (upright) that consists of 30 recordings. One subset is recorded as "ambient" and the other is recorded as "close" condition. We did not use other MAPS dataset for training our NN module. The model trained from thirty-five pieces of SMD was used for this test.

In this experiment, the evaluation is made only with the estimated distribution from the NN module μ_{nn} and σ_{nn} and ground truth μ_{gt} and σ_{gt}. Since the mapping between key velocity and MIDI velocity in SMD and the MAPS dataset is different, we cannot compare these values directly. Also, we cannot figure out how the same key velocity will be recorded as MIDI velocity in SMD and MAPS or which velocity value will make most close reproduction of a note in MAPS with the instrument in SMD. What we can assure is that MIDI velocity ranking of notes or piece will be preserved both in SMD and MAPS. Therefore we examine the Spearman correlation between the NN's guess μ_{nn} and σ_{nn} and the ground truth MAPS MIDI value μ_{gt} and σ_{gt}.

5.3 Procedure

The experiment procedure is as follows. First, the NMF module calculates note intensity and reproduces note-separated spectrograms for each pieces in the training set and test set. Then, we train the NN module with the note spectrograms of the training set from SMD. After the training, the trained NN estimates the velocity of note spectrograms of the test set. Combining the distribution of estimated velocity from the NN and estimated intensity from the NMF as described in section 4.3, we can obtain final MIDI velocity for each note in the piece. For the Experiment II, the calibration part is omitted. During the experiment, we used STFT with window size 8192, hop size 2048, and 8 spectral bases per pitch in the NMF module.

6. RESULTS

6.1 Experiment I: SMD

We present our result on the SMD set recorded in 2011 on Table 1. The ground truth velocity distribution of each piece is represented as GT, and the estimated distribution from the NN module is as NN. The remaining columns on the right are the average errors of four different mapping parameter for the same NMF result. UB1 is the first upper boundary that uses other test pieces to obtain the velocity-to-intensity mapping as in [15]. UB2 is the second upper boundary that assumes our NN module estimated the correct μ_V and σ_V. The proposed method (Prop.) is from the NN estimation for μ_V and σ_V. The baseline (Base) always guessed $\mu_V = 57.87$ and $\sigma_V = 16.25$. The last column shows the error when we directly used the NN estimation in note level, instead of combining it with the NMF intensity.

The estimation of the NN module showed high error in a note level as shown in the NN column. We presume the reason for the error is mainly based on the imperfection of source-separation. Also, the different recording condition in the test set could make not only intensity difference but also timbral change. This inhomogeneity may also have had a negative impact on the performance of the NN module.

Even though the note-level accuracy was not reliable, we found that the overall distribution of the estimated velocity resembles the distribution of ground truth velocity as we expected. By employing the estimated velocity distribution, the note intensity from the NMF module could be successfully mapped into MIDI velocity as shown in the Prop. column. The proposed system outperforms the baseline estimation in most pieces. While the fixed guess ignored characteristic of each piece, the NN module successfully estimated a correct distribution from the note spectrograms.

The difference between two upper boundary UB1 and UB2 shows the error caused by the assumption that the

Composers	Piece	Ground Truth		NN Estimation		UB1	UB2	Proposed	Baseline	NN note
		Mean	STD	Mean	STD	Err	Err	Err	Err	Err
Bach	BWV 888-1	49.7	12.6	53.3	15.5	3.1	3.9	3.3	6.6	10.4
Bach	BWV 888-2	63.3	11.3	62.8	13.3	2.1	3.1	3.5	9.0	10.1
Bartok	op. 80-1	68.9	18.2	65.7	15.3	5.9	6.6	8.5	15.0	12.2
Bartok	op. 80-2	59.5	23.5	59.5	20.4	5.1	7.2	8.6	10.3	11.3
Bartok	op. 80-3	67.4	19.0	64.8	17.3	6.0	7.1	8.9	14.8	13.0
Brahms	op. 5-1	64.8	23.5	62.0	19.6	7.2	8.4	10.0	13.1	13.8
Haydn	HobXVI-52-01	57.9	14.6	58.4	14.7	3.9	5.5	4.6	6.1	11.9
Haydn	HobXVI-52-2	49.8	18.6	53.9	16.6	3.8	4.7	5.6	8.0	11.1
Haydn	HobXVI-52-3	60.4	12.9	59.1	15.5	3.6	5.4	5.5	7.6	12.9
Mozart	K. 265	57.5	13.2	57.1	14.4	3.2	6.2	6.2	6.7	10.5
Mozart	K. 398	58.6	13.2	57.7	16.5	3.6	5.6	8.5	8.6	11.2
Rachmaninoff	op. 36-1	56.5	18.7	54.5	16.9	6.4	6.1	6.9	5.9	11.7
Rachmaninoff	op. 36-2	54.7	19.5	50.2	18.1	5.2	5.5	6.4	6.9	11.5
Rachmaninoff	op. 36-3	66.3	19.8	66.4	16.0	6.6	9.0	8.5	14.7	12.7
Ravel	Jeux d'eau	55.3	17.0	57.8	17.6	5.8	5.5	5.0	5.1	12.5
	Average					4.83	5.9	6.7	9.2	11.8

Table 1. The result of experiment on SMD. The first two columns show mean and standard deviation of note velocities from the ground truth and the estimation by neural network. "Err" stands for absolute mean error of note velocities. UB1 is an oracle model that learns key-dependent velocity mapping from other test pieces, and UB2 is another oracle model with ground-truth velocity mean and variance. The baseline model uses a global mean and variance. NN note represents mean error of velocity estimation of individual notes in the neural network

intensity-to-velocity mapping is consistent over the key. However, previous works showed that a piano stroke makes different intensity with the same velocity depending on the key [20]. This suggests the need of additional methods to compensate the key-dependent mapping in the future research.

The error is notable in *Rachmaninoff's Op. 36-1*. A possible reason is that the global setting of velocity distribution in the baseline is closer to the ground truth compared to the NN estimation. The errors in *Ravel's Jeux d'eau* is worth mentioning since the two upper boundary methods made the worse result. We presume that the reason is the frequent use of soft pedal during the performance. Soft pedal makes intensity lower, thus making our system estimate it softer than what is expected from its MIDI velocity.

6.2 Experiment II: MAPS

Figure 4 shows the correlation between the estimation from the NN module and the ground truth on the MAPS recordings. The absolute value of μ_{nn} and μ_{gt} has an error because of different key velocity to MIDI velocity mapping, thus cannot be compared directly. However, we can see that as the ground truth velocity mean of the piece increases, the estimated mean of NN also tends to catch it up. The same tendency is also found in the standard deviation. The Spearman correlation between μ_{GT} and μ_{NN} is 0.838, and that between σ_{GT} and σ_{NN} is 0.597.

Figure 4 also shows that the estimation from the NN module is not affected much by whether the recording is ambient or close, indicating that our NN module is robust to different piano and recording conditions. We did not apply the baseline method to MAPS because the estimation would be always constant regardless of the piece.

7. CONCLUSIONS

We presented a system that estimates key velocity from polyphonic piano recordings. The main limitation of previous work was the lack of method for calibration between intensity and key velocity. To overcome the limitation, We proposed a neural network module that takes note-separated spectrogram and estimates the key velocity of each note. Though the accuracy of individual notes is not reliable, the overall distribution resembles the distribution of ground truth velocity for each piece. Our system obtains a proper intensity-to-velocity mapping by employing the estimated velocity distribution, and then estimate the key velocity.

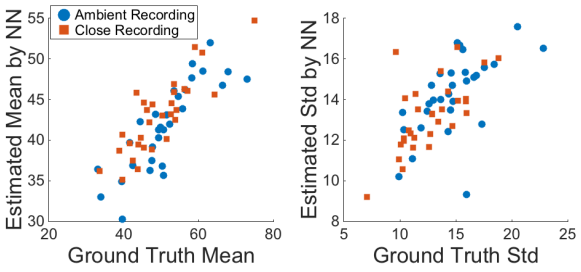

Figure 4. The test result on the MAPS dataset (Experiment II). Each point represents a single piece.

We evaluated our system on two different datasets. Overall, the evaluation showed a promising result of this timbre-based approach. The velocity estimation from the NN module showed a similar distribution with the ground truth velocity distribution despite the different recording conditions. Employing this estimated distribution, our system mapped note intensity to MIDI velocity reliably. Also, the result showed that our NN module learns robust features that can be applied to unseen data.

For the future work, we plan to apply our solution to real-world recordings with various timbre and recording conditions and, by combining other AMT and audio-to-score alignment algorithms, and obtain more full-fledged performance transcription.

8. ACKNOWLEDGEMENTS

This research was supported/partially supported by Samsung Research Funding & Incubation Center for Future Research.

9. REFERENCES

[1] Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri. Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems*, 41(3):407–434, 2013.

[2] D. G. Bhalke, C. B. Rama Rao, and D. S. Bormane. Automatic musical instrument classification using fractional fourier transform based- MFCC features and counter propagation neural network. *Journal of Intelligent Information Systems*, 46(3):425–446, Jun 2016.

[3] Roger B Dannenberg. The interpretation of MIDI velocity. In *Proc. of International Computer Music Conference (ICMC)*, pages 193–196, 1996.

[4] Patrick Joseph Donnelly et al. *Learning spectral filters for single-and multi-label classification of musical instruments*. PhD thesis, Montana State University-Bozeman, College of Engineering, 2015.

[5] Valentin Emiya, Roland Badeau, and Bertrand David. Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6):1643–1654, 2010.

[6] Antti Eronen and Anssi Klapuri. Musical instrument recognition using cepstral coefficients and temporal features. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages II753–II756, 2000.

[7] Sebastian Ewert and Meinard Müller. Estimating note intensities in music recordings. In *Proc. of 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 385–388, 2011.

[8] Sebastian Ewert and Meinard Müller. Using score-informed constraints for NMF-based source separation. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 129–132, 2012.

[9] Sebastian Ewert, Siying Wang, Meinard Müller, and M Sandler. Score-informed identification of missing and extra notes in piano recordings. In *Proc. of International Society of Music Information Retrieval Conference (ISMIR)*, pages 30–36, 2016.

[10] Werner Goebl and Roberto Bresin. Measurement and reproduction accuracy of computer-controlled grand pianos. *The Journal of the Acoustical Society of America*, 114(4):2273–2283, 2003.

[11] Yoonchang Han, Jaehun Kim, and Kyogu Lee. Deep convolutional neural networks for predominant instrument recognition in polyphonic music. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 25(1):208–221, 2017.

[12] Harry C Hart, Melville W Fuller, and Walter S Lusby. A precision study of piano touch and tone. *The Journal of the Acoustical Society of America*, 6(2):80–94, 1934.

[13] Toni Heittola, Anssi Klapuri, and Tuomas Virtanen. Musical instrument recognition in polyphonic audio using source-filter model for sound separation. In *Proc. of International Society for Music Information Retrieval Conference (ISMIR)*, pages 327–332, 2009.

[14] Kristoffer Jensen. *Timbre models of musical sounds*. PhD thesis, Department of Computer Science, University of Copenhagen, 1999.

[15] Dasaem Jeong and Juhan Nam. Note intensity estimation of piano recordings by score-informed NMF. In *Proc. of Audio Engineering Society Semantic Audio Conference*, 2017.

[16] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *Computing Research Repository*, abs/1412.6980, 2014.

[17] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. In *Advances in Neural Information Processing Systems*, pages 972–981, 2017.

[18] Meinard Müller, Verena Konz, Wolfgang Bogler, and Vlora Arifi-Müller. Saarland music data (SMD). In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR): Late Breaking session*, 2011.

[19] Bruno H Repp. Music as motion: A synopsis of Alexander Truslit's (1938) Gestaltung und Bewegung in der Musik. *Psychology of Music*, 21(1):48–72, 1993.

[20] Bruno H Repp. Some empirical observations on sound level properties of recorded piano tones. *The Journal of the Acoustical Society of America*, 93(2):1136–1144, 1993.

[21] Bruno H Repp. The dynamics of expressive piano performance: Schumann's ''Träumerei''revisited. *The Journal of the Acoustical Society of America*, 100(1):641–650, 1996.

[22] Craig Stuart Sapp. Comparative analysis of multiple musical performances. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 497–500, 2007.

[23] Wai Man Szeto and Kin Hong Wong. Source separation and analysis of piano music signals using instrument-specific sinusoidal model. In *Proc. of 16th International Conference on Digital Audio Effects (DAFx)*, 2013.

[24] Siying Wang, Sebastian Ewert, and Simon Dixon. 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, 2017.

[25] Gerhard Widmer, Simon Dixon, Werner Goebl, Elias Pampalk, and Asmir Tobudic. In search of the Horowitz factor. *AI Magazine*, 24(3):111, 2003.