PIANO SUSTAIN-PEDAL DETECTION USING CONVOLUTIONAL NEURAL NETWORKS

Beici Liang, György Fazekas and Mark Sandler

Centre for Digital Music, Queen Mary University of London, United Kingdom
{beici.liang, g.fazekas, mark.sandler}@qmul.ac.uk

ABSTRACT

Recent research on piano transcription has focused primarily on note events. Very few studies have investigated pedalling techniques, which form an important aspect of expressive piano music performance. In this paper, we propose a novel method for piano sustain-pedal detection based on Convolutional Neural Networks (CNN). Inspired by different acoustic characteristics at the start (pedal onset) versus during the pedalled segment, two binary classifiers are trained separately to learn both temporal dependencies and timbral features using CNN. Their outputs are fused in order to decide whether a portion in a piano recording is played with the sustain pedal. The proposed architecture and our detection system are assessed using a dataset with frame-wise pedal on/off annotations. An average F1 score of 0.74 is obtained for the test set. The method performs better on pieces of Romantic-era composers, who intended to deliver more colours to the piano sound through pedalling techniques.

***Index Terms*—** Piano sustain pedal, convolutional neural networks, playing technique detection.

1. INTRODUCTION

The sustain pedal is frequently used for seamless legato playing as well as the enrichment of sound in expressive piano performance. This is achieved using the piano mechanism whereby all dampers are lifted off the strings when the sustain pedal is pressed. Strings associated with the sounding notes are therefore sustained, while the others are slightly co-excited through sympathetic resonance. It is noted that pedalling techniques are not always indicated in music scores and can be played in many different ways, even if pedal markings are provided [1]. Automatic sustain-pedal detection can help reveal the secrets of artistic expressions in virtuoso performance. In this paper, we define piano pedalling detection as a task to localise the portions played with the sustain pedal in an audio recording.

This work is supported by Centre for Doctoral Training in Media and Arts Technology (EPSRC and AHRC Grant EP/L01632X/1), the EPSRC Grant EP/L019981/1 "*Fusing Audio and Semantic Technologies for Intelligent Music Production and Consumption (FAST-IMPACt)*" and the European Commission H2020 research and innovation grant AudioCommons (688382). Beici Liang is funded by the China Scholarship Council (CSC).

Existing studies on piano sustain-pedal detection take advantages of machine learning classifiers such as Support Vector Machines (SVM), combined with hand-crafted audio features of the sustain-pedal effects. In [2], notes played with or without the sustain pedal are separated using a threshold learnt through auto-regressive modelling of the energy of the residuals after the harmonics have been removed. The residual energy was found to increase when the sustain pedal is fully engaged in [3]. More features based on both harmonics and residuals were exploited to identify notes played with pedalling techniques of different timing and depth using a trained decision-tree-based SVM model in [4]. Since these features were extracted from isolated notes, dedicated new features were developed to detect pedalling in polyphonic music. A method for detecting onset times of legato pedalling (pressing the sustain pedal immediately after the note onset) was first proposed in [5] based on a measure of sympathetic resonance. Since a piano transcription technique was used as an intermediate step, the robustness of this method may be reduced by note transcription errors.

To overcome the limitations in previous works, we propose a novel detection method based on CNN using a new dataset. This consists of pieces by various composers recorded in pairs *with* and *without* pedals, see Sec. 3.1 for details. CNNs have been widely used to boost the performance in music information retrieval (MIR) tasks, such as tempo estimation [6], singing voice detection [7] and so on. In our approach, two CNN models with five 2D convolution layers were first trained for binary classification from pre-segmented fixed-length excerpts of pedal onsets and pedalled segments separately. They were then used as detectors in short-time analysis using overlapping windows to obtain local information in the pieces with various lengths. Their outputs were fused for identifying the presence/absence of the sustain pedal from every frame. To the best of our knowledge, this approach is the first to achieve piano sustain-pedal detection from polyphonic piano music. It can also be incorporated into a system for full transcription of piano music with the help of the state-of-the-art note event detection proposed in [8]. Python code of the experiments, along with the trained models, are made available online[1].

[1] https://github.com/beiciliang/sustain-pedal-detection

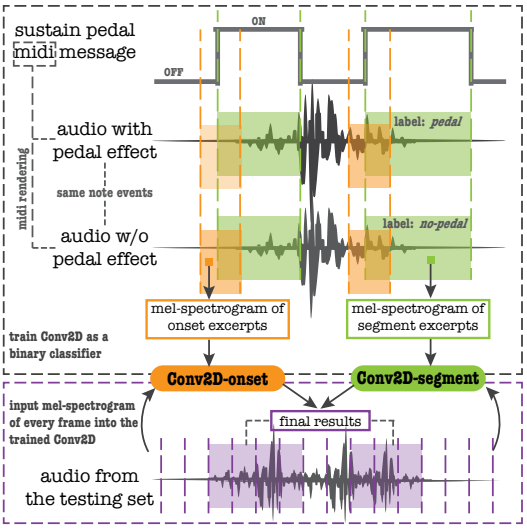

Fig. 1. Framework of the sustain-pedal detection system.

2. METHOD

Given that *pedal* and *no-pedal* versions of audio files with the same note events are available, two CNNs (see Sec. 2.1) are trained to predict whether a segment is played with or without pedal. Our approach, including how the models are trained, is illustrated in Figure 1. According to the ground truth encoded in MIDI data, onset and offset times of each sustain-pedal use can be obtained. These times are used to prepare excerpts in pairs (*pedal* and *no-pedal*) to train our CNN models as binary classifiers (collectively designated as `Conv2D`). This training strategy can facilitate the learning of features that are invariant to note event changes. To obtain results for each piece, we applied decision fusion [9] to the outputs of the classifiers `Conv2D-onset` and `Conv2D-segment` from sliding windows over the test piece in order to localise portions played with the sustain pedal.

2.1. Conv2D Training

According to piano acoustics and the observations in our prior works [4] and [5], musical features can be different in the frames around a pedal onset versus the ones within a pedalled segment. Here, a pedalled segment is referred to as samples between a pedal onset time and its corresponding offset time. Transient appears in the evolution of residual amplitude at the moment when the sustain pedal is pressed. Along with the engagement of the sustain pedal, more amplitude beatings are observed in the harmonics. The extent of sympathetic resonance and blurring effects are also enhanced. Therefore, we trained two models to distinguish excerpts with pedal onset and pedalled segment respectively from their associated *no-pedal* excerpts.

With polyphonic piano music, musical features related to the sustain pedal are rather subtle compared with features designed for conventional transcription tasks, such as note onset detection and pitch estimation. Designing hand-crafted features to represent the subtlety can be inefficient and may require computationally expensive algorithms. Inspired by the success of CNN in audio tagging [10], we applied CNN to learn discriminative features from log-amplitude mel-spectrograms. This input representation has been shown to yield significant improvements in the music tagging task [11].

input of `Conv2D-onset` or `Conv2D-segment`
mel-spectrogram (#frequency × #time × #channel)
convolution
(3 × 20 × 7) ; (20 × 3 × 7) ; (3 × 3 × 7)
max-pooling (2 × 2)
convolution (3 × 3 × 21)
max-pooling (2 × 2)
convolution (3 × 3 × 21)
max-pooling (2 × 2)
convolution (3 × 3 × 21)
max-pooling (4 × 4)
fully-connected (2 units)
output (softmax)

Table 1. Proposed structure and configurations of `Conv2D`.

We used identical architectures to train the two `Conv2D` models as described in Table 1. It was proposed in [12] that using different musically motivated filter shapes in the first layer of CNN could achieve higher prediction accuracy. As we discussed above, pressing the sustain pedal can result in both temporal and spectral changes. Therefore, shapes of the filters in the first layer were set to $3 \times 20 \times 7$, $20 \times 3 \times 7$ and $3 \times 3 \times 7$, which correspond to dimensions: frequency $\times$ time $\times$ channel. Thereby larger time/frequency contexts can be modelled. The outputs of the first convolution layers were concatenated together along the channel dimension. The following three layers each have $3 \times 3 \times 21$ filters. At each convolution layer, we applied zero padding such that the output has the same length as the input. Other configurations and hyper-parameters were configured to prevent the network from over-fitting and to accelerate convergence as follows. Batch Normalisation was added after every convolution. The output of every convolutional layer was then passed through a Rectified Linear Unit (ReLU), followed by a max-pooling layer and a dropout layer with the probability 0.25 to aid generalisation [13]. The final layer is fully-connected with average-pooling and softmax activation in order to map the output to the range [0,1]. This can be interpreted as a likelihood score of the presence of the sustain pedal in an excerpt.

With the above architecture, two `Conv2D` models were trained using the Adam optimiser [14] to minimise categorical cross entropy. Both obtained better performance than other models in the binary classification tasks, which are discussed in Sec. 3.2. Since our main focus is to analyse the performance of the proposed method on sustain-pedal detection, for the brevity of this paper, effects of different configurations and hyper-parameters are not discussed.

2.2. Fusion of `Conv2D-onset` and `Conv2D-segment`

Our database includes piano pieces of different lengths. The detectors `Conv2D-onset` and `Conv2D-segment` are first used separately. Their outputs from short-time sliding windows over the mel-spectrogram of a piece were thresholded at 0.98 to obtain binary decisions at a higher precision. Detection was then reinforced by decreasing the rate of false positives through fusion, as described by Algorithm 1. Let D_{ons} and D_{seg} be the lists of binary decisions produced by the two detectors, T_{ons} and T_{seg} be the associated lists of frame times in second. D is the list of final detection result that implies the sustain pedal is *on* or *off* at every frame. According to the ground-truth annotation, we can evaluate the performance of our detection method considering all the frames of the test set as discussed in Sec. 3.3.

Algorithm 1 Decision fusion

Require: τ: the tolerance time window
  procedure FUSION($D_{ons}, D_{seg}, T_{ons}, T_{seg}$)
    $D \leftarrow zeros(D_{seg})$
    for all $j \in \{1, ..., len(D_{seg}) - 2\}$ **do**
      if $D_{seg}[j-1] \wedge D_{seg}[j] \wedge D_{seg}[j+1]$ **then**
        for all $i \in \{0, ..., len(D_{ons}) - 1\}$ **do**
          if $D_{ons}[i] \wedge abs(T_{ons}[i] - T_{seg}[j-1]) < \tau$ **then**
            $D[j-1, j, j+1] \leftarrow D_{seg}[j-1, j, j+1]$
    return D

3. EXPERIMENT AND RESULTS

3.1. Dataset

Since we were not able to find a dataset including both *pedal* and *no-pedal* versions of audio files, we decided to create a dataset large enough to train the `Conv2D`. For this purpose, 1567 MIDI files publicly available from the Minnesota International Piano-e-Competition website[2] were downloaded. They were recorded using a Yamaha Disklavier which can capture nuances from the performance of skilled competitors. Pianoteq 6 PRO[3], a physically modelled virtual instrument approved by Steinway & Sons, was used to render high quality audio with a sampling rate of 44.1 kHz and a resolution of 24 bits from these MIDI files. We employed the Steinway Model D grand piano instrument and close-miking recording mode using a pair of figure-of-eight U87 microphones. Audio with and without sustain-pedal effect was then generated through preserving or removing the sustain-pedal message in the MIDI data. We used audio data generated from the year 2011 Competition as the test set, which covers pieces by 28 different composers from Baroque to the Modern period. Data from other years of the competition were shuffled to form the training and validation set. We obtained 1113/279/175 pieces, i.e., 70%/20%/10% split for training/validation/testing.

[2]http://www.piano-e-competition.com

[3]https://www.pianoteq.com/pianoteq6

Model	Onset	Segment
`MFCC-SVM`	0.8471	0.9173
`Conv2D-3x3`	0.9791	0.9965
`Conv2D`	**0.9852**	**0.9972**

Table 2. Best AUC-ROC scores of the three models.

Ground-truth annotations for each piece consist of binary labels (*on* and *off*) indicating whether the sustain pedal is pressed or released at every frame. They were obtained by thresholding the sustain-pedal MIDI message in range [0,127] at 64. A pedal onset is determined to have happened during a frame where the pedal state changes from *off* to *on*. A pedalled segment is determined to start at a pedal onset and finish when the state returns to *off*. According to the distribution of pedalled-segment duration calculated from all the MIDI files, the sustain pedal is commonly pressed for between 0.3 and 2.3 seconds. To prepare fixed-length excerpts for training `Conv2D-onset`, we choose 0.5-second excerpts around every pedal onset. Excerpts for training `Conv2D-segment` were clipped from pedalled segments which are more than 0.3-second long and then processed to obtain 2 seconds in length[4]. The start and end times of these *pedal* excerpts were also used to obtain *no-pedal* excerpts from audio without sustain-pedal effect. Therefore excerpts were arranged in pairs. The training/validation set contains 893062/241670 excerpts for `Conv2D-onset` and 707944/195454 excerpts for `Conv2D-segment`. Mel-spectrograms with 128 mel-bands were extracted from excerpts in real-time on the GPU using *Kapre* [15], which can simplify audio preprocessing and save storage. Time-frequency transformation was performed using 1024-point FFT with a hope size of 441 samples (10 ms). We used *Keras* [16] and *Tensorflow* [17] frameworks in our implementation.

3.2. Binary Classification

To examine whether the proposed CNN architecture can better discriminate *pedal* versus *no-pedal* excerpts through modelling larger time/frequency contexts, `Conv2D` was compared with another two models (namely `Conv2D-3x3` and `MFCC-SVM`) in binary classification tasks. `Conv2D-3x3` replaced the first convolution layer of `Conv2D` by $3 \times 3 \times 21$ filters, which was originally proposed for image classification [18] and has been found to be effective in music classification [19]. It captures fewer temporal and spectral dependencies than `Conv2D`. `MFCC-SVM` was inspired by methods in conventional MIR tasks, which involve hand-crafted features and machine learning classifiers. It used the means and standard deviations of 20 Mel-Frequency Cepstral Coefficients (MFCCs), and their first and second-order derivatives as features to train the SVM. We used *Librosa* [20] for MFCC extraction, and *Scikit-learn* [21] for SVM construction.

[4]Pedalled segments that are shorter/longer than 2 seconds are repeated/trimmed to create a 2-second excerpt.

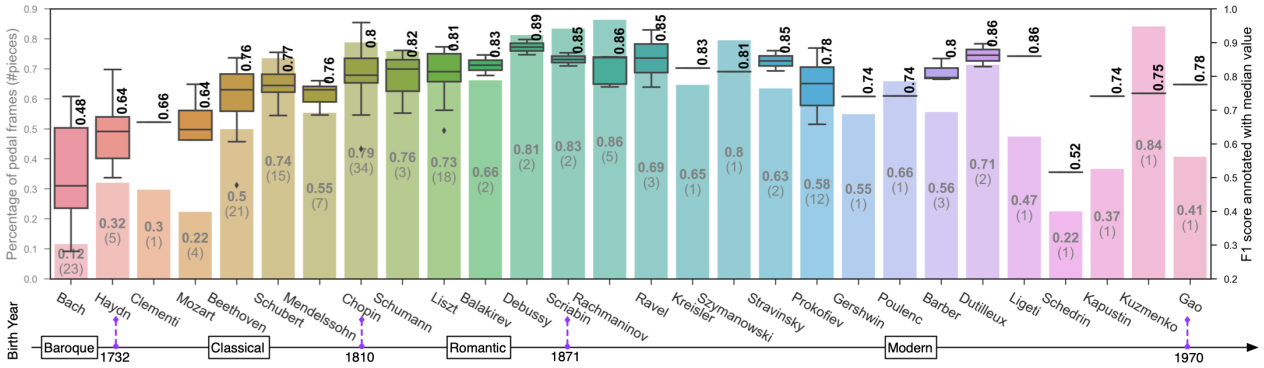

Fig. 2. Box plot of F1 score and bar plot of *pedal*-frame proportion ordered by composer's lifetime.

Both `Conv2D` and `Conv2D-3x3` were trained until the validation accuracy no longer improved for 10 epochs. Batch size was set to 250 examples randomly selected from the training set. The SVM parameters were optimised using grid-search based on the validation results. Both linear and radial kernels were used. Bandwidth for radial kernel was selected from range $[1/2^3, 1/\textit{feature vector dimension}]$. The penalty parameter was selected from rage [0.1, 32]. Considering the large size of our dataset, roughly $1/10$ of the training/validation set, i.e., 70000/20000 from excerpts of pedal onset and pedalled segment separately, were used for comparing the models. In the binary classification tasks of identifying excerpts with pedal onset or pedalled segment, the three models' best AUC-ROC scores (Area Under Curve - Receiver Operating Characteristic) based on the validation set are presented in Table 2. `Conv2D` achieved the highest scores in both tasks. This demonstrates the better performance of our proposed model. We further trained `Conv2D-onset` and `Conv2D-segment` with the whole training set. We can obtain accuracy scores of 0.9470 and 0.9901 respectively for the entire validation set.

3.3. Detection from Polyphonic Music

We applied sliding windows to a test piece in order to get decision outputs from the two trained models separately at every frame. The windows for `Conv2D-onset` and `Conv2D-segment` cover a duration of 0.5 and 0.3 seconds, with a hope size equivalent to 0.01 and 0.1 seconds respectively. In particular, the 0.3-second samples were tiled to two seconds such that the input size was coherent with the one in the training phase. Following the decision fusion policy introduced in Sec. 2.2, we first located portions that had more than three frames continuously considered as *pedal* by `Conv2D-segment`. If `Conv2D-onset` also returned *pedal* within 0.1 second around the beginning of a portion, the sustain pedal was detected as *on* in the frames of this portion. The rest of the frames were assigned to *off*. We finally obtained frame-wise *on*/*off* results for a piece.

Our detection method was evaluated on every piece in the test set using four common evaluation measures. The accuracy is the proportion of frames correctly labelled. Precision, recall and F1 score are calculated with respect to label *on*. There are 435999 *on* and 282795 *off* frames in total. From this we obtain average values of the four measures: 0.7964, 0.8572, 0.6655 and 0.7422. A detailed look at the F1 scores for the pieces by different composers is presented as a box plot with median value annotation in Figure 2. The percentage of the *on* frames according to the ground truth and the number of pieces associated to each composer are also shown. Our detection method inclines towards pedalled frames. It works best for the pieces around the Romantic era, when modern pedalling techniques appear to have been established and became widely used. The highest F1 median value of 0.89 was obtained for Debussy's pieces. For pieces that rely less on the sustain pedal in performance, such as the ones in the Baroque and early-Classical era, detection performance could be influenced by the increasing false positive rate.

4. CONCLUSION

In this paper, we presented a new approach for piano sustain-pedal detection. We took advantage of CNNs to model the temporal and spectral contexts within the first layer with different filter shapes. We can thus capture the nuances of two phases of pressing the sustain pedal. Two corresponding models with the same architecture were trained as binary classifiers using excerpts in pairs (*pedal* versus *no-pedal*). Their decision outputs were fused to locate segments played with the pedal from polyphonic music. Our experimental results show that this method is useful for indicating onset and offset times of the sustain pedal, which are essential for interpreting most of the classical piano pieces. Considering our dedicated dataset, the reduced acoustic complexity may lead to generalisation issues on commercial recordings. We believe our trained models can be efficiently adapted to various real-world scenarios using transfer learning. This consists our future work.

5. REFERENCES

[1] Sandra P Rosenblum, "Pedaling the piano: A brief survey from the eighteenth century to present," *Performance Practice Review*, vol. 6, no. 2, pp. 8, 1993.

[2] Roland Badeau, Nancy Bertin, Bertrand David, Antony Schutz, and Dirk Slock, "Piano "forte pedal" analysis and detection," in *124th Convention of the Audio Engineering Society*, 2008.

[3] Heidi-Maria Lehtonen, Henri Penttinen, Jukka Rauhala, and Vesa Välimäki, "Analysis and modeling of piano sustain-pedal effects," *The Journal of the Acoustical Society of America*, vol. 122, no. 3, pp. 1787–1797, 2007.

[4] Beici Liang, György Fazekas, and Mark B Sandler, "Detection of piano pedaling techniques on the sustain pedal," in *143rd Convention of the Audio Engineering Society*, 2017.

[5] Beici Liang, György Fazekas, and Mark B Sandler, "Piano legato-pedal onset detection based on a sympathetic resonance measure," in *26th European Signal Processing Conference (EUSIPCO)*, 2018.

[6] Hendrik Schreiber and Meinard Müller, "A single-step approach to musical tempo estimation using a convolutional neural network," in *19th International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 98–105.

[7] Kyungyun Lee, Keunwoo Choi, and Juhan Nam, "Revisiting singing voice detection: A quantitative review and the future outlook," in *19th International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 506–513.

[8] Curtis Hawthorne, Erich Elsen, Jialin Song, Adam Roberts, Ian Simon, Colin Raffel, Jesse Engel, Sageev Oore, and Douglas Eck, "Onsets and frames: Dual-objective piano transcription," in *19th International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 50–57.

[9] David L Hall and James Llinas, "An introduction to multisensor data fusion," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 6–23, 1997.

[10] Eduardo Fonseca, Manoj Plakal, Frederic Font, Daniel P. W. Ellis, Xavier Favory, Jordi Pons, and Xavier Serra, "General-purpose tagging of freesound audio with audioset labels: Task description, dataset, and baseline," Submitted to DCASE2018 Workshop, 2018.

[11] Keunwoo Choi, György Fazekas, Kyunghyun Cho, and Mark B Sandler, "A comparison on audio signal preprocessing methods for deep neural networks on music tagging," in *26th European Signal Processing Conference (EUSIPCO)*, 2018.

[12] Jordi Pons and Xavier Serra, "Designing efficient architectures for modeling temporal features with convolutional neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2472–2476.

[13] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[14] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations (ICLR)*, 2015.

[15] Keunwoo Choi, Deokjin Joo, and Juho Kim, "Kapre: On-gpu audio preprocessing layers for a quick implementation of deep neural network models with keras," in *Machine Learning for Music Discovery Workshop at 34th International Conference on Machine Learning*. ICML, 2017.

[16] François Chollet et al., "Keras," `https://keras.io`, 2015.

[17] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, Software available from tensorflow.org.

[18] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations (ICLR)*, 2014.

[19] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho, "Convolutional recurrent neural networks for music classification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 2392–2396.

[20] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th Python in Science Conference (SciPy)*, 2015, pp. 18–25.

[21] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, et al., "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.