Transfer Learning for Piano Sustain-Pedal Detection

Beici Liang, György Fazekas and Mark Sandler
Centre for Digital Music, Queen Mary University of London
London, United Kingdom
Email: {beici.liang,g.fazekas,mark.sandler}@qmul.ac.uk

Abstract—Detecting piano pedalling techniques in polyphonic music remains a challenging task in music information retrieval. While other piano-related tasks, such as pitch estimation and onset detection, have seen improvement through applying deep learning methods, little work has been done to develop deep learning models to detect playing techniques. In this paper, we propose a transfer learning approach for the detection of sustain-pedal techniques, which are commonly used by pianists to enrich the sound. In the source task, a convolutional neural network (CNN) is trained for learning spectral and temporal contexts when the sustain pedal is pressed using a large dataset generated by a physical modelling virtual instrument. The CNN is designed and experimented through exploiting the knowledge of piano acoustics and physics. This can achieve an accuracy score of 0.98 in the validation results. In the target task, the knowledge learned from the synthesised data can be transferred to detect the sustain pedal in acoustic piano recordings. A concatenated feature vector using the activations of the trained convolutional layers is extracted from the recordings and classified into frame-wise pedal press or release. We demonstrate the effectiveness of our method in acoustic piano recordings of Chopin’s music. From the cross-validation results, the proposed transfer learning method achieves an average F-measure of 0.89 and an overall performance of 0.98 in the validation results. In the target task, the knowledge acquired from the synthesised data can be used to detect the sustain pedal, including music score, melspectrogram and messages from MIDI or sensor data.

In this paper, we focus on detecting the technique of the sustain pedal, which is the most frequently used one among the three standard piano pedals. All dampers are lifted off the strings when the sustain pedal is pressed. This mechanism helps to sustain the current sounding notes and allows strings associated to other notes to vibrate due to coupling via the bridge. A phenomenon known as sympathetic resonance [1] is thereby enhanced and embraced by pianists to create a “dreamy” sound effect. We can observe how the phenomenon reflects on the melspectrogram in Figure 1, where note F4 is played without (first) and with (second) the sustain pedal in two bars respectively. Note that the symbol under the second bar of the music score in Figure 1 can be used to indicate the sustain-pedal techniques. Yet, even if pedal notations are provided, pedalling in the same piano passage can be executed in many different ways. Playing techniques are typically adjusted to the performer’s sense of tempo, dynamics, as well as the location where the performance takes place [2].

Given that detecting pedalling nuances from the audio signal alone is a rather challenging task [3], several measurement systems have been developed to capture the pedal movement. For instance, the Yamaha Disklavier piano can encode this movement into MIDI messages (0-127) along with note events. However, their effects are typically obscured by the variations in pitch, dynamics and other elements in polyphonic music. Therefore, automatic detection of pedalling techniques using hand-crafted features is a challenging problem. Given enough labelled data, deep learning models have shown the ability of learning hierarchical features. If these features are able to represent acoustic characteristics corresponding to pedalling techniques, the model can serve as a detector.

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).

Fig. 1. Different representations of the same note played without (first note) or with (second note) the sustain pedal, including music score, melspectrogram and messages from MIDI or sensor data.
not practical. We approach the sustain-pedal detection from
the audio domain using transfer learning [5] as illustrated
in Figure 2. Transfer learning exploits the knowledge gained
during training on a source task and applies this to a target
task [6]. This is crucial for our case, where the target-task data
is obtained from recordings of a different piano, therefore it
is difficult to learn a “good” representation due to mechanical
and acoustical deviations. In our source task, a convolutional
neural network (denoted by $\text{convnet}$ hereafter) is trained
for distinguishing synthesised music excerpts with or without
the sustain-pedal effect. The $\text{convnet}$ is then used as a
feature extractor, aiming to transfer the sustain-pedal effect
learned from the source task to the target task. Support
vector machines (SVMs) [7] are trained using the frame-
wise $\text{convnet}$ features from the acoustic piano recordings
to finalise the feature representation transfer as the target task.
SVMs can be used as a classifier to localise which frames are
played with the sustain pedal. The performance is expected to
improve significantly with the new feature representation. To
sum up, the main contributions of this paper are:

1) A novel strategy of model design, which incorporates
knowledge of piano acoustics and physics, enabling the
$\text{convnet}$ to become more effective in representing the
sustain-pedal effect.

2) A transfer learning method that allows the $\text{convnet}$
trained from the source task to be adapted to the target
task, where the recording instruments and room acous-
tics are different. This also allows effective learning with
a smaller dataset.

3) Finally, we conduct visual analysis on the convolutional
layers of the $\text{convnet}$ to promote model designs with
fewer trainable parameters, while maintaining their dis-
crminating power.

The rest of this paper is organised as follows. We first
introduce related works in Section II. The process of database
construction is described in Section III. The methods of
sustain-pedal detection including $\text{convnet}$ design and trans-
fer learning are discussed in Section IV. Experiments and
results are presented in Section V. We finally conclude our
work in Section VI.

II. RELATED WORK

Past research in music information retrieval (MIR) abound
in recognition of musical instruments, but automatic detection
of instrumental playing techniques (IPT) remains underde-
veloped [8]. IPT creates a variety of spectral and temporal
variations of the sounds in different instruments. Recent re-
search has attempted to transcribe IPT on drum [9], erhu
[10], guitar [11], [12] and violin [13], [14]. Hand-crafted
features are commonly designed based on instrument acoustics
to capture the salient variations induced by IPT. The sustain-
pedal technique leads to rather subtle variations, therefore most
studies managed to detect the technique based on isolated
notes only [15]–[17]. This challenge is further intensified in
polyphonic music where clean features extracted from isolated
notes cannot be easily obtained. In our prior work [18], the first
research aiming to extract pedalling technique in polyphonic
piano music, we proposed a method for detecting pedal onset
times using a measure of sympathetic resonance. Yet, this
method assumes the availability of modelling the specific
acoustic piano which is also used in evaluation. Moreover,
it is prone to errors due to its reliance on note transcription.

Convolutional Neural Networks (CNNs) have been used
to boost the performance in MIR tasks, with the ability to
efficiently model temporal features [19] and timbre representa-
tions [20]. We choose CNNs to facilitate learning time-
frequency contexts related to the sustain pedal, using syn-
thesised excerpts in pairs (pedal versus no-pedal versions).
Using this method, contexts that are invariant to large pitch
dynamics changes can be learned.

To apply a $\text{convnet}$ trained from the synthesised data into
the context of real recordings, a transfer learning approach
can be used. It has been gaining more attentions in MIR for
alleviating the data sparsity problem and its ability to be used
for different tasks. For example, Choi et al. [21] obtained
features from CNNs, which were trained for music tagging in
the source task. These features outperformed MFCC features
in the target tasks, such as genre and vocal/non-vocal classi-
fication. We believe such strategy is suited to the challenges
in detecting the sustain pedal from polyphonic piano music
recorded in different acoustic and recording conditions.

In our case, training a $\text{convnet}$ with the synthesised data
is considered as the source task. Then in the target task, we
can use the learnt representations from the trained $\text{convnet}$
as features, which are extracted from every frame of a real
piano recording, to train a dedicated classifier adapted to the
actual acoustics of the piano and the performance venue used
in the recording. This transfer learning approach is expected
to better identify frames played with the sustain pedal. For the
dedicated classifier in the target task, we opt for SVM instead
of multi-layer perceptron because SVM can greatly reduce the
training time and yield better generalisation in classification
tasks [22]. In Section V-B, compared with fine-tuning the last
layer of the pre-trained $\text{convnet}$, transfer learning with SVM
trained using the activations of multiple layers also achieves
better performance.
III. DATASET

For the source task, pedal and no-pedal versions of music excerpts are required to train a convnet, which is able to highlight the spectral or temporal characteristics that change with the sustain pedal instead of note events. For this reason, 1392 MIDI files publicly available from the Minnesota International Piano-e-Competition website\(^1\) were downloaded. They were recorded using a Yamaha Disklavier piano from the performance of skilled competitors. To render these MIDI files into high quality audio, the Pianoteq 6 PRO\(^2\) software was used. This physically modelled virtual instrument approved by Steinway & Sons can export audio using models of different instruments and recording conditions. We employed the Steinway Model D grand piano instrument and the close-miking recording mode. Audio with or without sustain-pedal effect was then generated with a sampling rate of 44.1 kHz and a resolution of 24 bits. These were rendered while preserving or removing the sustain-pedal message in the MIDI data. For each pedal-version audio, we can obtain the temporal regions when the sustain pedal is on or off by thresholding the MIDI message at 64 given its range of [0,127]. A pedalled segment is determined to start at a pedal onset (where the pedal state changes from off to on) and finish when the state returns to off. We can clip all the pedalled segments to form the pedal excerpts. The start and end times of the pedalled segments were also used to obtain no-pedal excerpts from the corresponding no-pedal-version of the audio.

These music excerpts were derived from pieces by 84 different composers from Baroque to the Modern period. Their durations distribute between 0.3 and 2.3 seconds. To prepare fixed-length data for training, excerpts that are shorter or longer than 2 seconds were repeated or trimmed to create a 2-second excerpt. Considering the large size of our dataset, we randomly took a thousand samples from the excerpts of each composer. In total, 62424 excerpts form a smaller dataset\(^3\). This also helps to compare convnet of different architectures in a more efficient way, since the training time can be significantly reduced.

For the target task, the dataset consists of ten well known passages of Chopin’s piano music. A pianist was asked to perform the passages using a Yamaha baby grand piano situated in the MAT studios at Queen Mary University of London. The audio were recorded at 44.1 kHz and 24 bits using the spaced-pair stereo microphone technique with a pair of Earthworks QTC40 omnidirectional condenser microphones positioned about 50 cm above the strings. The positions were kept constant during the recording. Meanwhile, movement of the sustain pedal was recorded along with the audio with the help of the measurement system proposed in [4]. The audio data were annotated with frame-wise on or off labels as the ground truth, representing whether the sustain pedal was pressed or released in each audio frame. The occurrence counts of the labels in each passage are presented in Table III. It can be observed that the sustain pedal was frequently used for the interpretation of Chopin’s music.

IV. METHOD

A. CNN for binary classification

Given our large training data consisting of excerpts arranged in pedal/no-pedal pairs, binary classification was chosen as a source task. This enabled the convnet to focus on variations in the nuances on sound played with/without the sustain pedal, while invariant to other musical elements such as pitch and loudness. Considering that the use of the sustain pedal can have effects on every piano string, this could lead to changes that affect the entire spectrum, i.e., take place at a global level. Therefore representations that reveal finer details, such as short-time Fourier transform (STFT), may become inefficient for training. The mel-spectrogram is a 2D representation that approximates human auditory perception through aggregating STFT bins along the frequency axis. This computationally efficient input has been shown to be successful in MIR tasks such as music tagging [23]. For the above reasons, we consider mel-spectrogram an adequate input representation.

Inspired by Vggnet [24] which has been found to be effective in music classification [25], our convnet model uses a similar architecture with fewer trainable parameters to learn the differences in time-frequency patterns in pedal versus no-pedal cases. The model consists of a series of convolutional and max-pooling layers, which are followed by one fully-connected layer with two softmax outputs. The architecture we propose to start with, and the related hyperparameters are summarised in Figure 3, where \((c, (m, n))\) correspond to \((\text{channel}, (\text{kernel lengths in frequency, time}))\) specifying the convolutional layers. Pooling layer is specified by \((\text{pooling length in frequency, time})\).

It was noted in [20] that designing filter shapes within the first layer can be motivated by domain knowledge in order to efficiently learn musically relevant time-frequency contexts.

---

\(^1\)http://www.piano-e-competition.com  
\(^2\)https://www.pianoteq.com/pianoteq6  
\(^3\)There are less than a thousand excerpts for some of the composers. The excerpts were sampled in pairs such that the ratio of pedal and no-pedal excerpts is 1:1.
with spectrogram-based CNNs. To decide \((m, n)\) of the first layer yielding the best representational power, we selected their values motivated by piano acoustics and physics, which can substantially change the sustain-pedal effect. Performance of \texttt{convnet} with different filter shapes within the first layer were evaluated using the validation set as discussed in Section V-A. Apart from the common small-square filter shape, the shapes we experimented with are either wider rectangles in the time domain to model short time-scale patterns, or in the frequency domain to fit spectral contexts.

In every convolutional layer, batch normalisation was used to accelerate convergence. The output was then passed through a Rectified Linear Unit (ReLU) [26], followed by a max-pooling layer to prevent the network from over-fitting and to be invariant to small shifts in time and frequency. To further minimise over-fitting, global average pooling was used before the final fully-connected layer. The final layer used softmax minimisation in order to map the output to the range \([0,1]\), which can be interpreted as a likelihood score of the presence of activation in order to map the output to the range \([0,1]\). The final layer used softmax to accelerate convergence. The output was then passed through a Rectified Linear Unit (ReLU) [26], followed by a max-pooling layer to prevent the network from over-fitting and to be invariant to small shifts in time and frequency. To further minimise over-fitting, global average pooling was used before the final fully-connected layer. The final layer used softmax activation in order to map the output to the range \([0,1]\), which can be interpreted as a likelihood score of the presence of the sustain pedal in the input. We trained \texttt{convnet} with the Adam optimiser [27] to minimise binary cross entropy.

There are possibilities that simpler model architecture, i.e., with fewer channels or convolutional layers, would be sufficient for our binary classification task using reduced parameters. We explored the effect of number of channels and layers in Section V-A. The best performing \texttt{convnet} model was selected to ensure the features extracted from it can accurately represent the acoustic effects when the sustain pedal is used.

B. Transfer Learning

When the detection aims at real piano recordings, relying on the output from the trained \texttt{convnet} may be inadequate. This is because our \texttt{convnet} was trained solely on synthesised excerpts in pairs. Only the hierarchical features representing acoustic characteristics when the sustain pedal of a virtual piano is played in the specified recording environment can be learned. It has been well understood that piano sounds can be varied by brands, and also affected by room acoustics and recording conditions. Such differences could bring more variations to the sustain-pedal effect. These serve as motivations for the proposed transfer learning, which could extract the hierarchical knowledge (specialised features) from the \texttt{convnet}. The knowledge is then used as features to train a dedicated classifier for detecting the sustain pedal of a specific piano in real scenarios.

The activations of each intermediate layers were subsampled using average pooling and then concatenated into the final \texttt{convnet} features as demonstrated in Figure 3. Here average pooling can summarise the global statistics and reduce the size of feature maps to a vector of length associated to the value of \(c\). In the end, a \(c \times 4\) dimensional feature vector was generated since there are 4 convolutional layers in the \texttt{convnet}. For the brevity of this paper, the effects of using various strategies for layer-wise feature combination are not discussed.

To identify which audio frames were played with the sustain pedal, we can use SVMs to classify the frame-wise \texttt{convnet} features into pedal on or off states. SVMs were chosen first because we assume the features extracted from the carefully-trained model in the source task should be representative and separable. Second, the SVM algorithm was originally devised for classification problems, involving finding the maximum margin hyperplane that separates two classes of data and has been shown ideal for such a task [28]. This allows us to focus on the quality of learnt features. SVMs were trained using a supervised learning method in the target task, where the SVMs were trained using a supervised learning method in the target task, where the detection was done on acoustic piano recordings.

As shown in Section V-B, the proposed transfer learning method overall outperformed the case of using the pre-trained \texttt{convnet} output directly. It also provided better performance than using the pre-trained \texttt{convnet} with a fine-tuned last layer, which is a common approach to transfer learning.

V. Experiment

In our experiments, melspectrograms with 128 mel bands were extracted from excerpts to serve as input to the network. The processing was done in real-time on the GPU using Kapre [29], which can simplify audio preprocessing and saves storage. Time-frequency transformation was performed using 1024-point FFT with a hop size of 441 samples (10 ms). Keras [30] and Tensorflow [31] frameworks were used for the implementation.

A. Source Task

The 62424 excerpts were split into 80%/20% to form the training/validation set. Models were trained until the validation accuracy no longer improved for 10 epochs. Batch size was set to 128 examples. To examine which \texttt{convnet} model can

---

### Table I

| Model                        | \((m, n)\) | Accuracy | AUC   |
|------------------------------|------------|----------|-------|
| \texttt{convnet-baseline}    | (3, 3)     | 0.9755   | 0.9963|
| \texttt{convnet-frequency}   | (9, 3)     | 0.9630   | 0.9905|
| \texttt{convnet-multi}       | (45, 3)    | 0.9747   | 0.9988|
| \texttt{convnet-time}        | (3, 10)    | 0.9815   | 0.9971|

### Table II

| Models with Reduced Parameters | \(c\) | \(l\) | Accuracy | AUC   |
|--------------------------------|------|------|----------|-------|
| \texttt{convnet-multi}         | 3    | 2    | 0.8781   | 0.9486|
|                                | 12   | 2    | 0.9389   | 0.9904|
|                                | 21   | 2    | 0.9552   | 0.9980|
|                                | 31   | 2    | 0.9436   | 0.9949|
|                                | 41   | 2    | 0.9708   | 0.9949|
|                                | 51   | 2    | 0.9764   | 0.9949|
|                                | 61   | 2    | 0.9840   | 0.9949|
|                                | 71   | 2    | 0.9870   | 0.9949|
|                                | 81   | 2    | 0.9935   | 0.9949|
|                                | 91   | 2    | 0.9702   | 0.9949|

| Original Model                 | 21   | 4    | 0.9837   | 0.9983|

Note: \(l\) denotes the number of convolutional layers.
### TABLE III
PERFORMANCE OF THE TWO METHODS IN THE TARGET TASK.

| Music Passages | Occurrence Counts | Retrain Last Layer Only | Transfer Learning with SVM |
|----------------|-------------------|-------------------------|----------------------------|
|                | Retrain | Last Layer Only | Retrain | Last Layer Only | Retrain | Last Layer Only | Retrain | Last Layer Only |
| Op.10 No.3     | 849    | 268              | 0.7615  | 0.9965          | 0.8633  | 0.8457          | 0.9941  | 0.9139          |
| Op.23 No.1     | 722    | 355              | 0.6670  | 0.8575          | 0.7505  | 0.8645          | 0.9349  | 0.8982          |
| Op.28 No.4     | 905    | 322              | 0.7509  | 0.9698          | 0.8502  | 0.8148          | 0.9839  | 0.8922          |
| Op.28 No.6     | 788    | 280              | 0.7357  | 0.9607          | 0.8332  | 0.8178          | 0.9569  | 0.8819          |
| Op.28 No.12    | 297    | 66               | 0.7321  | 0.9566          | 0.8523  | 0.8097          | 0.9385  | 0.8668          |
| Op.28 No.15    | 413    | 306              | 0.7659  | 0.9149          | 0.7771  | 0.8412          | 0.9624  | 0.8997          |
| Op.28 No.20    | 785    | 274              | 0.7405  | 0.9449          | 0.8400  | 0.7640          | 0.9974  | 0.8780          |
| B.39            | 660    | 197              | 0.7720  | 0.9439          | 0.8949  | 0.8025          | 0.9419  | 0.9432          |
| Op.28 No.20    | 591    | 180              | 0.7622  | 0.9272          | 0.8666  | 0.9640          | 0.9702  | 0.8688          |
| Average         | 740    | 270              | 0.7392  | 0.9387          | 0.8262  | 0.8591          | 0.9396  | 0.8938          |

**Fig. 4.** Overall performance of the three methods in the target task.

The number of channels ($c$) was set to 21 for all convolutional layers. Table I presents the accuracy and AUC scores of the above models obtained from the validation set. According to the best AUC score of convnet-frequency and convnet-time respectively, we selected (45,3) and (3,10) along with (3,3) to create another model with multiple filter shapes (convnet-multi). To be specific, the first convolutional layer of convnet-multi consisted of (7, (45,3)), (7, (3,10)) and (7, (3,3)) producing outputs, as we concatenated along the channel dimension. The best accuracy and AUC scores were achieved by convnet-multi, i.e., 0.9837 and 0.9983.

It is noted that all the models above obtained AUC score higher than 0.99 due to the relative simplicity of the classification task. To examine if the same level of performance can be obtained with fewer trainable parameters, we trained models similar to convnet-multi but with fewer channels and convolutional layers. According to the results in Table II, the original convnet-multi remains the model with the highest score of AUC. Therefore, it was selected as the final model from the source task in order to be used as a feature extractor in the following target task.

**B. Target Task**

In the target task, sliding window was applied to the acoustic piano recordings in order to extract features of the trained convnet-multi model at every frame, as introduced in Section IV-B. The window covers a duration of 0.3 seconds.
with a hop size equivalent to 0.1 seconds. The 0.3-second samples were then tiled to 2 seconds and transformed into melspectrogram such that the input size was coherent with the one in the source task. The extracted features were used to train the SVM constructed by Scikit-learn [32].

The experiment was done by conducting leave-one-group-out cross-validation, where samples were grouped in terms of music passages. The performance of the proposed transfer learning method was validated in each music passage where the frame-wise features need to be classified by the SVM into pedal on or off, while the rest of the passages constitute the training set. The SVM parameters were optimised using grid-search based on the validation results. Radial kernel was used. Its bandwidth and the penalty parameter were selected from the ranges below:

- bandwidth: [1/2^3, 1/2^5, 1/2^7, 1/2^9, 1/2^11, 1/2^13, 1/feature vector dimension]
- penalty parameter: [0.1, 2.0, 8.0, 32.0]

We compared the proposed transfer learning method with the detection using a fine-tuned convnet-multi model, which can serve as a baseline classifier. Here “fine-tuning” is referred to as only retraining the fully-connected layer of convnet-multi. This is commonly considered a basic transfer learning technique. Within each cross-validation fold, the fully-connected layer was updated until the accuracy stopped increasing for 10 epochs. Then we obtained the fine-tuned convnet-multi outputs from short-time sliding windows over the melspectrogram of the validation passage.

Given the frame-wise on/off results for every music passage, we calculated precision ($P_1$), recall ($R_1$) and F-measure ($F_1$) with respect to the label on. They are defined as:

$$P_1 = \frac{N_{tp}}{N_{tp} + N_{fp}}, \quad R_1 = \frac{N_{tp}}{N_{tp} + N_{fn}}, \quad F_1 = 2 \times \frac{P_1 \times R_1}{P_1 + R_1},$$

where $N_{tp}$, $N_{fp}$ and $N_{fn}$ are the numbers of true positives, false positives and false negatives respectively.

Table III presents the performance measurement of the two methods respectively for every validation passage in the cross-validation fold, where the occurrence counts of label on and off were obtained from the ground truth. In general, our proposed transfer learning method with SVM obtains better performance. We can observe that the average value of $P_1$ and $F_1$ are 11.99% and 6.76% higher in using the transfer learning method with SVM than with the fine-tuned convnet-multi. Both methods achieved similar average value of $R_1$.

We also compared the overall performance of the two methods along with directly using the pre-trained convnet-multi. Their results are presented passage by passage in Figure 4. Considering the imbalanced occurrence counts of the two labels, the micro-averaged F-measure ($F_{\text{micro}}$) was selected to evaluate the overall performance, because it calculates metrics globally by counting the total $N_{tp}$, $N_{fp}$ and $N_{fn}$ with respect to both labels. The proposed transfer learning method with SVM presents the best overall performance with more than 10% higher than the $F_{\text{micro}}$ obtained by the other two methods.

C. Discussion

To gain deeper insight into the pros and cons of our method, we visualised the detection results of the last 15 seconds in the passage of Op.66, which obtained the best performance in the target task. Figure 5 highlights the audio frames corresponding to pedal on according to the detection results and the ground truth separately. Most of the frames were correctly identified. Yet, there were false positives because some frames prior to the true sustain-pedal onset times were detected as positives, hence $P_1$ was decreased. This implies a model dedicated to the detection of the sustain-pedal onset should be developed. Such model itself or its outputs could be fused with convnet-multi in order to localise the pedalled segments with a better precision. There was also fragmentation corresponding to transient on returned by the model, leading to increasing $N_{fp}$ and $N_{fn}$. This can be reduced by post-processing techniques.

It is notable in the source task that the AUC of models with various filter shapes within the first layer obtained scores that were all higher than 0.99 as shown in Table I. We assume that pressing the sustain pedal could result in acoustic characteristics that significantly change the patterns in both frequency and time. Thereby convnet-multi can obtain the highest AUC score. To understand the learning process of the convnet models, we conducted a visual analysis of the deconvolved melspectrogram of music excerpts in pairs, which have the same note event, but differently labelled. Visualisation results using the convnet-frequency and convnet-time with the best AUC score, i.e., with $(m, n)$ set to (45, 3) and (3, 10), are shown in Figure 6a and Figure 6b respectively. In Figure 6, we only select the first feature maps separately learned in the four convolutional layers and present their deconvolved melspectrograms. From layer 1 to 3, the two models both focus on the time-frequency contexts centred around the fundamental frequency and their partials. More contexts in the higher frequency bands can be learned by the convnet-frequency. In the fourth layer, only the first half of melspectrograms are emphasised. We could infer the sustain pedal has more effects on the notes which the pedal just started to play together with.

![Fig. 5. Visualisation of the ground truth (top row) and the detection result (bottom row) in Op.66. Audio frames that are annotated/detected as pedal on are highlighted in orange/green.](image)
differences between *pedal* and *no-pedal* excerpts lie in the lower frequency bands indicated by the *convnet-time*. Considering that a slightly lower accuracy score was obtained by *convnet-frequency*, we can assume dependencies within the higher frequency range could be a redundant knowledge to learn in our source task.

Another observation is that performance of the binary classification task is less dependent on the effect of the number of layers, according to the scores in Table II. This also reflects in the changes shown by the deconvolved melspectrograms from layer 1 to 3, where roughly the same time-frequency areas were emphasised. We could train *convnet* models in a more efficient way, using fewer convolutional layers, while keeping or increasing the number of channels.

Through inspection of the detection results and the learned filters, we can extend our understanding of the *convnet* models in music. This inspires us to develop CNN models not only for detecting the pedalled frames, but also for learning the transients introduced by the sustain-pedal onset or even the offsets. More audio data including pieces by other composers and using various recording conditions should be tested to verify the robustness of our approach. This also constitutes our future works.

**VI. CONCLUSION**

In this paper, we answered the question: “Can a computer point out pedalling techniques when a piano recording from a virtuoso performance is given?”. A novel transfer learning approach based on *convnet* models was proposed to detect the sustain pedal, and evaluated on ten passages of Chopin’s music. A specific transfer learning paradigm was used where the source and target tasks differ in objectives and experimental conditions, including the use of synthesised versus real acoustic recordings. The model trained in the source task can then be employed as a feature extractor in the target task.
In the source task, the model architecture was informed by piano acoustics and physics in order to facilitate the training process. Given the synthesised excerpts played with or without the sustain pedal, we showed that convnet models can learn the time-frequency contexts corresponding to acoustic characteristics of the sustain pedal, instead of larger variations introduced by other musical elements. Among all models, convnet-multi was selected to be used in the target task due to its highest scores of accuracy and AUC in binary classification. Features with more representation power dedicated to the sustain-pedal effect can be extracted from the intermediate layers of convnet-multi. This helps to adapt the detection to acoustic piano recordings. Thus a better performance measurement was obtained compared to fine-tuning or directly applying the pre-trained convnet-multi network. Finally, visualisation of the learned filters using deconvolution showed us potential directions towards designing more efficient and effective models for detecting different phases of the use of the sustain pedal.

REFERENCES

[1] C. Morley, Dictionary of acoustics Academic Press. Academic Press, 2001.
[2] S. P. Rosenblum, “Pedaling the piano: A brief survey from the eighteenth century to the present,” Performance Practice Review, vol. 6, no. 2, pp. 158–178, 1993.
[3] W. Goebi, S. Dixon, G. De Poli, A. Friberg, R. Bresin, and G. Widmer, “Sense in expressive music performance: Data acquisition, computational studies, and models,” Sound to Sense - Sense to Sound: A State of the Art in Sound and Music Computing, pp. 195–242, 2008.
[4] B. Liang, G. Fazekas, and M. Sandler, “Measurement, recognition, and visualization of piano pedaling gestures and techniques,” Journal of the Audio Engineering Society, vol. 66, no. 6, pp. 448–456, 2018.
[5] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, Deep learning. MIT press Cambridge, 2016, vol. 1.
[6] S. J. Pan and Q. Yang, “A survey on transfer learning,” IEEE Transactions on knowledge and data engineering, vol. 22, no. 10, pp. 1345–1359, 2010.
[7] J. A. Suykens and J. Vandewalle, “Least squares support vector machine classifiers,” Neural processing letters, vol. 9, no. 3, pp. 293–300, 1999.
[8] V. Lounetlen, J. Andén, and M. Lagrange, “Extended playing techniques: The next milestone in musical instrument recognition,” in Proceedings of the 5th International Workshop on Digital Libraries for Musicology (DL/M), 2018.
[9] C.-W. Wu, C. Dittmar, C. Southall, R. Vogl, G. Widmer, J. Hockman, M. Muller, and A. Lerch, “A review of automatic drum transcription,” IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), vol. 26, no. 9, pp. 1457–1483, 2018.
[10] L. Yang, K. Z. Rajab, and E. Chew, “The filter diagonalisation method for music signal analysis: frame-wise vibrato detection and estimation,” Journal of Mathematics and Music, pp. 1–19, 2017.
[11] L. Su, L.-F. Yu, and Y.-H. Yang, “Sparse cepstral, phase codes for guitar playing technique classification,” in Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR), 2014, pp. 9–14.
[12] Y.-P. Chen, L. Su, and Y.-H. Yang, “Electric guitar playing technique detection in real-world recording based on f0 sequence pattern recognition,” in Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR), 2015, pp. 708–714.
[13] P.-C. Li, L. Su, Y.-h. Yang, A. W. Su et al., “Analysis of expressive musical terms in violin using score-informed and expression-based audio features,” in Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR), 2015, pp. 809–815.
[14] A. Perez-Carrillo and M. M. Wanderley, “Indirect acquisition of violin instrumental controls from audio signal with hidden markov models,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 23, no. 5, pp. 932–940, 2015.