PITCH-INFORMED INSTRUMENT ASSIGNMENT USING A DEEP CONVOLUTIONAL NETWORK WITH MULTIPLE KERNEL SHAPES

Carlos Lordelo[1,2] **Emmanouil Benetos**[1] **Simon Dixon**[1] **Sven Ahlbäck**[2]
[1] Queen Mary University of London, UK [2] Doremir Music Research AB, Sweden
c.p.viannalordelo@qmul.ac.uk

ABSTRACT

This paper proposes a deep convolutional neural network for performing note-level instrument assignment. Given a polyphonic multi-instrumental music signal along with its ground truth or predicted notes, the objective is to assign an instrumental source for each note. This problem is addressed as a pitch-informed classification task where each note is analysed individually. We also propose to utilise several kernel shapes in the convolutional layers in order to facilitate learning of timbre-discriminative feature maps. Experiments on the MusicNet dataset using 7 instrument classes show that our approach is able to achieve an average F-score of 0.904 when the original multi-pitch annotations are used as the pitch information for the system, and that it also excels if the note information is provided using third-party multi-pitch estimation algorithms. We also include ablation studies investigating the effects of the use of multiple kernel shapes and comparing different input representations for the audio and the note-related information.

1. INTRODUCTION

Automatic Music Transcription (AMT) is the process of creating any form of notation for a music signal and is currently one of the most challenging and discussed topics in the Music Information Retrieval (MIR) community [1]. Most AMT systems are designed to transcribe a single monophonic or a single polyphonic source into a musical score (or piano-roll). In this case, the main sub-task involved in the process is Multi-Pitch Estimation (MPE), where predictions regarding the pitch and time localisation of the musical notes are carried out. However, when analysing polyphonic multi-instrumental recordings, not only each note should have its pitch and duration properly estimated, but the information regarding the timbre of sounds should also be correctly processed [2]. It is mandatory to have a way of recognising the instrument that played each note.

In this paper, we propose a pitch-informed instrument assignment approach, where the main objective is to associate each note event of a music signal to one instrument class. In contrast to other state-of-the-art instrument recognition approaches, which are usually addressed on a frame-level [3, 4] or clip-level [5, 6] basis, our approach analyses each note event individually. Therefore, it is possible to say that we perform a note-level instrument recognition.

Previous work has shown that the use of pitch information can help frame-level instrument recognition [3]. Inspired by this, we propose a framework that uses an auxiliary input based on note-event pitch information. Our system is trained using the note annotations provided in the MusicNet [7] dataset. However, our main motivation is to create a modular framework that can be combined with any MPE algorithm in order to obtain multi-instrumental pitch predictions, which allows for transcribing music in staff notation, corresponding to the perception of pitch events. Therefore, we also show that our approach can obtain good performance when the note information is predicted by state-of-the-art MPE algorithms such as [8, 9].

Furthermore, the utilisation of multiple kernel shapes in the filters of a Convolutional Neural Network (CNN) has been proven to be an efficient strategy of applying domain knowledge in several MIR tasks [10–12]. In particular, [12] applied this strategy with a dense connectivity pattern of skip-connections in order to learn even more efficient feature maps and reduce the number of trainable parameters for the task of source separation. In our work, we build our CNN adapting the architecture in [12] for the classification (instrument assignment) task and verified that it can also improve its performance. In summary, the main contributions of this paper are as follows:

- Pitch-informed instrument assignment: Proposal of a Deep Neural Network (DNN) that associates each note from a music signal to its instrumental source.

- Modular Framework: Approach works with any MPE method. We evaluate the performance when using ground-truth note labels as well as 2 state-of-the-art MPE algorithms [8, 9].

- Multiple Kernel Shapes: Proposal of a CNN architecture for instrument assignment that uses multiple kernel shapes for the convolutions, facilitating learning representations for different instruments and note sound states. We show that their use improves instrument assignment performance.

Set		Piano	Violin	Viola	Cello	Horn	Bassoon	Clarinet	Harps.	Bass	Oboe	Flute	Total
Train		628549	197229	88446	89356	10770	13874	22873	4914	3006	8624	8310	1075951
		58.4%	18.3%	8.2%	8.3%	1.0%	1.2%	2.1%	0.5%	0.3%	0.8%	0.8%	100%
Test		5049	3238	842	1753	557	873	1277	0	0	0	0	13589
		37.2%	23.8%	6.2%	12.9%	4.1%	6.4%	9.4%	0	0	0	0	100%

Table 1. Statistics of the note labels provided by MusicNet across train and test sets.

2. RELATED WORK

The instrument recognition task is usually formulated as a multi-label classification task that can be addressed either on a frame-level [3, 4, 13], where the purpose is to obtain the instrument activations across time, or on a clip-level basis [5, 6, 14], where the purpose is to estimate the instruments that are present in an audio clip. However, our objective in this work is to approach the instrument recognition task note by note, assigning an instrument class to each. Such a task requires note-event annotations and, in the literature, it is also known as *instrument assignment* [15] or *multi-pitch streaming* [2].

Just few works have explored this particular task. For instance, Duan et al. [2] approached it using a constrained clustering of frame-level pitch estimates obtained from an MPE algorithm via the minimisation of timbre inconsistency within each cluster. They tested different timbre features for both music and speech signals. In [16], a similar method was proposed, where the authors applied Probabilistic Latent Component Analysis (PLCA) to decompose the audio signal into multi-pitch estimates and to extract source-specific features. Then, clustering was performed under the constraint of cognitive grouping of continuous pitch contours and segregation of simultaneous pitches into different source streams using Hidden Markov Random Fields. Both of those works, however, assume that each source is monophonic, i.e., each instrument could only play a single note at a time.

An alternative approach iss to model the temporal evolution of musical tones [15]. This method is based around the use of multiple spectral templates per pitch and instrument source that correspond to sound states. The authors used hidden Markov model-based temporal constraints to control the order of the templates and streamed the pitches via shift-invariant PLCA. In a more recent work, Tanaka et al. [17] also approached the task via clustering, but applied on a joint input representation combined of the spectrogram and the pitchgram, which was obtained using an MPE algorithm. In their proposal, each bin of the joint input was encoded onto a spherical latent space taking into account timbral characteristics and the piano-rolls of each instrument were later estimated via masking of the pitchgram based on the results of a deep spherical clustering technique applied on the latent space.

Recent multitask deep-learning based works have successfully proposed multi-instrumental AMT methods that are able to directly estimate pitches and associate them to their instrumental source jointly [4, 18, 19]. In [18], a multitask deep learning network jointly estimated outputs for various tasks including multiple-pitch, melody, vocal and bass line estimation. The Harmonic Constant-Q Transform (HCQT) of the audio signal was used as input and the data used for training was semi-automatically labelled by remixing a diverse set of multitrack audio data from the MedleyDB [20] dataset. In [4] a DNN was used to jointly predict the pitch and instrument for each audio frame. They used the Constant-Q Transform (CQT) as input to their system and trained using a large amount of audio signals synthesised from MIDI piano-rolls. Manilow et al. [19], on the other hand, were able to jointly transcribe and separate an audio signal into up to 4 instrumental sources — piano, guitar, bass and strings. However, their system was trained with only synthesised signals.

Our approach is closely related to that of Hung and Yang [3], where a frame-level instrument recogniser is proposed using the CQT spectrogram of the music signal allied with the pitch information of the note events. We also use the pitch annotations to guide the instrument classifier, but our work differs from [3] in the fact that we perform a classification for each note event individually, while Hung and Yang use the whole piano-roll at once to guide frame-level instrument recognition. While Hung and Yang are able to obtain the instrument activations leveraging from the pitch information, they cannot stream the note events into their corresponding instruments.

3. PROPOSED METHOD

In our method, we use the same definition of note events as in the MIREX MPE task [1]. Each note N is considered an event with a constant pitch f_0, an onset time T_{on} and an offset time T_{off}. Therefore, if a music signal has a total of M notes, any note N_i, with $i \in \{1, \cdots, M\}$, can be uniquely defined by the tuple $(f_0^i, T_{\text{on}}^i, T_{\text{off}}^i)$. In our experiments, we use two ways of obtaining this note information. The first using ground-truth pitch labels provided by the employed dataset (MusicNet) [7] and the second using pitch estimates predicted by state-of-the-art MPE algorithms [8, 9]. We consider the f_0 granularity to follow the semitone scale, ranging from $A0$ to $G\sharp 7$ (MIDI #21 − 104).

In our framework polyphony is allowed, so, most of the time more than a single note will be active, but our objective is to analyse each note of the audio signal separately in order to be able to assign an instrument class to it. This is done by using two inputs to the model: the main in-

[1] https://www.music-ir.org/mirex/

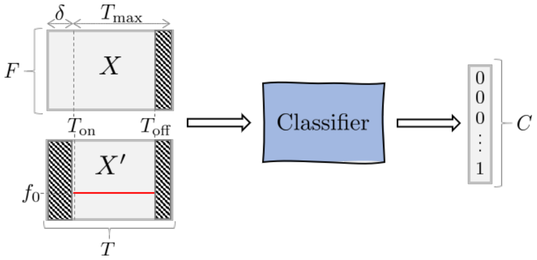

Figure 1. Overview of the proposed framework for instrument assignment. See Section 3 for the detailed explanation of the variables in the figure.

put $X(f,t)$, with f representing frequency and t representing time, is a time-frequency representation of a segment of the audio signal around the value of T_{on}, and an auxiliary input $X'(f,t)$, which carries information regarding f_0, T_{on} and T_{off}. The two inputs are concatenated into a two-channel input $\mathbf{X}(f,t,c)$, where $c \in \{1,2\}$ represents the channel dimension, that is fed to the model. In Figure 1 an overview of the proposed framework is shown.

3.1 Main Audio Input

The main input is a time-frequency representation $X(f,t) \in \mathbb{R}^{F \times T}$ of a small clip of the music signal, where F is the number of frequency bins and T is the number of time frames. The clip is generated by first setting a maximum duration $T_{\max}$ for the note. We tested values of $T_{\max}$ ranging from 400 ms to 1 s (see Section 7 for details) and 400 ms obtained the best results, so we kept this value in all of the other experiments. If any note N_i has a duration D_i greater than $T_{\max}$, i.e., $D_i = T_{\text{off}} - T_{\text{on}} > T_{\max}$, only its initial time span of $T_{\max}$ seconds is considered.

Next, for every note N_i, $X_i(f,t)$ is constructed by picking a segment of duration $T = T_{\max} + \delta$ from the original music signal starting from $T^i_{\text{on}} - \delta$, where δ is a small interval to take into account deviations between the true onset value and the value we use. The inclusion of the extra window of δ from the music signal also helps the convolutional layers since it brings some context of the signal before the note onset value. We set $\delta = 30$ ms after initial tests. Lastly, if the note duration D_i is less than $T_{\max}$, we set the values of $X(f, t > D_i)$ to zero, where $D_i = T^i_{\text{off}} - T^i_{\text{on}}$.

3.2 Auxiliary Note-Related Input

The auxiliary input $X'(f,t) \in \mathbb{R}^{F \times T}$ is a harmonic comb representation using the pitch value f_0 as the first harmonic [2], such that,

$$X'(f,t) = \begin{cases} 1, & \text{if } \ f = hf_0 \ \text{ and } \ T_{\text{on}} \le t \le T_{\text{off}} \\ 0, & \text{otherwise} \end{cases}, \tag{1}$$

where $h = \{1,2,3,\cdots,H\}$ with H being the total number of harmonics in the representation. We tested multiple values for H (see Section 6). In practice, we use a tolerance of half a semitone for each harmonic value when constructing $X'(f,t)$ as a mel-spectrogram. Therefore, even though this representation starts as binary, the final mel-spectrogram is not binary due to the mel-filtering procedure. Moreover, it is important to note that we also set the values of X' before T_{on} and after T_{off} to zero. In Figure 2 we show an example of a pair of inputs for our framework.

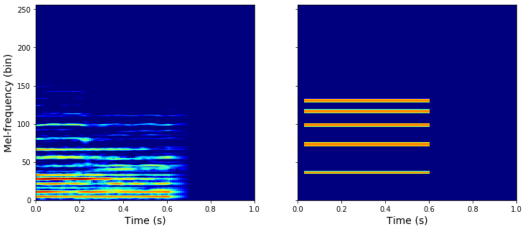

Figure 2. A pair of inputs using 256 mel-frequency spectrogram. In the left is depicted $X(f,t)$, where three pitches are simultaneously activated (MIDI # 58, 62 and 74) and in the right $X'(f,t)$, where the note with pitch # 74, is modelled using an harmonic comb of $H = 5$. In this example, $D_i = 600$ ms and $T_{\max} = 1$ s.

3.3 Output

The note-level instrument assignment task is tackled as a multi-class single-label classification task. Given $\mathbf{X}$, our objective is to classify it as belonging to one of C instrument classes. We use a deep neural network that receives $\mathbf{X}$ as input and outputs a C-dimensional vector $\hat{y}$. A softmax activation function is applied in the final layer of the network to ensure the values of $\hat{y}$ represent probabilities that sum up to 1. The model is trained using the cross-entropy loss. At inference time, the class corresponding to the dimension with the highest value in $\hat{y}$ is predicted. See Section 4 for details regarding the network architecture.

In the cases where two or more instruments are playing the same pitch simultaneously, the small differences between the notes' onset and offset values can generate different inputs $\mathbf{X}$. Thus, it would still allow the instrument assignment task to be properly executed as a single label classification scenario. However, when the pitch, onset and offset values of notes from different instruments exactly match, our system will consider them as a single note and only a single instrument will be estimated. This case rarely happens in real-world scenarios for many musical styles. For instance, in MusicNet only 0.9% of the notes had the same pitch, onset, and offset values. For our experiments, we have considered notes in MusicNet that were performed by a single instrument, and discarded the notes that were concurrently produced (in terms of the same pitch, onset, and offset times) by multiple instruments. As a proof of concept, we believe that this is not a severe limitation for our framework and we leave multi-labelled approaches as future work.

4. ARCHITECTURE

When processing music spectrograms by CNNs, the strategy of combining vertical and horizontal kernel shapes in

[2] We use the definition that f_0 corresponds to the first harmonic.

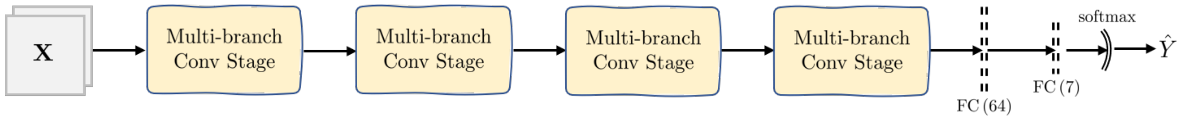

Figure 3. Proposed network architecture. FC represents a Fully Connected layer with LeakyRelu activation function.

Main Input	Aux Input	Piano	Violin	Viola	Cello	Horn	Bassoon	Clarinet	Mean
CQT	—	0.960	0.732	0.116	0.725	0.512	0.296	0.681	0.575
	$H = 1$	**0.994**	0.934	0.763	0.955	0.783	**0.888**	0.950	**0.895**
	$H = 2$	0.993	0.939	0.771	**0.960**	**0.785**	0.858	0.929	0.891
	$H = 3$	0.992	**0.946**	**0.772**	0.957	0.754	0.884	**0.952**	0.894
	$H = 4$	0.993	0.938	0.766	0.958	0.784	0.869	0.950	0.894
	$H = 5$	0.993	0.939	0.767	0.952	0.769	0.874	0.949	0.892
Mel STFT	—	0.967	0.742	0.222	0.730	0.607	0.306	0.690	0.609
	$H = 1$	0.996	0.939	0.759	0.958	0.780	0.867	0.958	0.895
	$H = 2$	0.994	0.945	0.779	0.956	0.809	0.864	0.946	0.899
	$H = 3$	**0.997**	0.944	0.775	**0.958**	0.8104	0.879	**0.967**	**0.904**
	$H = 4$	0.996	0.935	0.747	0.945	**0.839**	**0.891**	0.960	0.902
	$H = 5$	0.996	**0.947**	**0.783**	0.954	0.801	0.876	0.954	0.902

Table 2. Evaluation of instrument assignment task when using CQT or Mel spectrograms as input representation for the network as well as a comparison between models trained with no auxiliary input and models trained with different number of harmonics in the auxiliary input. This experiment was performed using $T_{max} = 400\,\text{ms}$

the model architecture can facilitate learning of timbre-discriminative feature maps [10–12]. In our work, we propose a CNN adapted from the 3W-MDenseNet [12]. This architecture was originally proposed for harmonic-percussive source separation and consists of an encoder-decoder model that estimates spectrograms for two sources. Thus, the outputs of the 3W-MDenseNet have the same shape as the mixture spectrogram that is used as input. In this CNN architecture, three MDenseNets [21] run in parallel in separated branches, each with a unique kernel shape (vertical, square and horizontal). The MDenseNets are only combined at the final layer, i.e., after both the encoding and decoding procedure are performed. In our work, we adopt a similar methodology by taking only the encoder layers from [12] and adding fully connected layers at the end in order to perform classification rather than separation. Also, we propose modifications to the original encoder layers: instead of combining the branches using a concatenation layer only at the final stage, we concatenate their feature maps at the end of each down-sampling stage. By doing so, we allow each branch to have access to feature maps computed using all different choices of kernel shapes from a previous stage.

Figure 3 shows a summary of the architecture we adopt in our work. It consists of a stack of 4 multi-branch convolutional stages and 2 fully connected layers. In Figure 4 the internal structure of the multi-branch convolutional stage is shown. Internally, each multi-branch convolutional stage contains 3 separate branches whose convolutions have unique kernel shapes. We use a branch with horizontal (1×9), a branch with square (3×3), and a branch with vertical (9×1) convolutions. In each path, a Densely connected convolutional Network (DenseNet) [22] with growth rate $k = 25$ and number of layers $L = 4$ is used. In short, a DenseNet is a stack of L k-channel convolutional layers — each with its own activation function — with a dense pattern of skip connections, where each layer receives the concatenation of all previous layers' outputs as input. We used the LeakyRelu function as the activation function for all layers. The reader is referred to [22] for the detailed internal structure of a DenseNet. After the DenseNet, a (2×2) max pooling layer is applied in order to reduce the feature maps' dimensions and increase the receptive field at each branch. Afterwards, the three branches are concatenated and the batch is normalised. The final feature maps are used as input for the next multi-branch convolutional stage. Since we need to concatenate feature maps that were originated by multiple kernel shapes we use padding on the convolution and on the max pooling to ensure the feature maps maintain the same dimensions across branches. The number of trainable parameters is approximately 1.1 million.

5. DATASET

We used the MusicNet dataset [7] in our experiments. MusicNet is the largest publicly available dataset with non-synthesised data that is strongly labelled for the task of

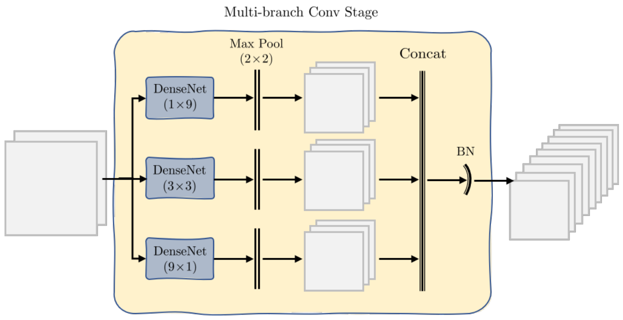

Figure 4. Internal structure of a multi-branch conv. stage. Each DenseNet has a growth rate $k = 25$ and $L = 4$ layers. BN represents a Batch Normalisation layer. We use LeakyRelu as activation function after each conv. layer.

instrument recognition. This means that we know the exact frames where the instruments are active in the signal, which permits the training of supervised models to perform instrument recognition at the frame-level, note-level, and clip-level. The dataset contains 330 freely-licensed classical music recordings by 10 composers, written for 11 instruments, along with over 1 million annotated labels indicating the precise time and pitch of each note in the recordings and the instrument that plays each note.

The instrument taxonomy of MusicNet is: piano, violin, viola, cello, french horn, bassoon, clarinet, harpsichord, bass, oboe and flute. However, the last 4 instruments (harpsichord, bass, oboe and flute) do not appear in the original test set provided by the authors. Therefore, in all our experiments we ignored all the labels related to those instruments and we performed a 7-class instrument classification using the following classes: piano, violin, viola, cello, french horn, bassoon and clarinet. Table 1 shows the statistics of the note labels provided by MusicNet. The dataset is heavily biased towards piano and violin given their usual presence in Western classical music recordings.

6. EXPERIMENTAL SETUP

In all experiments we used the original train/test split provided by MusicNet with the original sampling frequency of 44100 Hz. For experiments that involved the computation of Short Time Fourier Transform (STFT) we used Blackman-Harris windows of 4096 samples to compute the Discrete Fourier Transform (DFT). The hop size was always set to 10 ms in every experiment.

From the training set we picked 5% of the notes of each class and created a validation set. We trained the models using the Adam optimiser with an initial learning rate of 0.001 and reduced it by a factor of 0.2 if the cross-entropy loss stopped improving for 2 consecutive epochs on the validation set. If no improvement happened after 10 epochs, the training was stopped early. The experiments were performed using the Tensorflow/Keras Python package.

The classification performance was evaluated by computing the note-level F-score (F_s), which is directly related to the precision (P), recall (R) according to:

$$P = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}},\; R = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}},\; F_\mathrm{s} = \frac{2PR}{P+R} \qquad (2)$$

where TP is the number of true positives, FP the false positives and FN the false negatives.

For the cases when the instrument assignment is done on top of MPE algorithms, we provide 2 groups of metrics that are generated following the MIREX evaluation protocol for the music transcription task. In the first group, an estimated note is assumed correct if its onset time is within 50 ms of a reference note and its pitch is within quarter tone of the corresponding reference note. The offset values are ignored. In the second group, on top of those requirements, the offsets are also taken into consideration. An estimate note is only considered correct if it also has an offset value within 50 ms or within 20% of the reference note's duration around the original note's offset, wherever is largest. After all notes are verified, the F-score is computed note-wise across time and the average value is provided here. This evaluation method was computed using the `mir_eval.transcription`[3] toolbox.

7. RESULTS

7.1 Effects of the Kernel Shapes

First we analysed the effects of the inclusion of multiple kernel shapes in the architecture of the CNN. The top part of Table 3 compares 3 versions of the model: one that uses only square filters in a single branch; a version using the branched structure, but with (3×3) kernels in each; and another model with the proposed multi-branch structure with horizontal, square, and vertical kernels. For the single-branched case we increased the growth factor of the DenseNets to 57 channels in order to keep the number of trainable parameters of the network close to the original.

Analysing the results we see that the addition of new kernel shapes improved the average F-score across all classes. Regarding each instrument class, we can say that for string instruments (piano, violin, viola and cello) there is a gain in performance, while for non-string instruments (horn, bassoon, clarinet) the performance either drops or remains with a negligible gain if compared to the models that used only square filters. This suggests that the inclusion of vertical kernel shapes helped the model in learning the percussive characteristics of the timbre of string musical instruments.

7.2 Evaluation of the Input Size

We also tested different values for the input size. More specifically, we compared multiple values for T_max, which is the maximum valid window of analysis for a note event. The results are shown in the lower part of Table 3. We can see that the shortest input size of 400 ms obtained the best results. We believe that it is due to the fact that the average duration of a note event in the test set of MusicNet is

[3] `https://craffel.github.io/mir_eval/`

Kernel / $T_{\max}$	Piano	Violin	Viola	Cello	Horn	Bassoon	Clarinet	Mean
(3×3)	0.994	0.936	0.757	0.954	**0.826**	0.864	0.954	0.898
$3\times(3\times3)$	0.995	0.939	0.764	0.945	0.819	**0.896**	0.965	0.903
Multiple	**0.997**	**0.944**	**0.775**	**0.958**	0.810	0.879	**0.967**	**0.904**
400 ms	**0.997**	**0.944**	**0.775**	**0.958**	0.810	0.879	**0.967**	**0.904**
600 ms	0.996	0.942	0.771	0.954	**0.826**	**0.881**	0.959	0.904
800 ms	0.996	0.944	0.772	0.957	0.814	0.868	0.965	0.902
1 s	0.997	0.931	0.740	0.954	0.742	0.871	0.948	0.883

Table 3. Instrument assignment performance based on the kernel shapes used in network (first three rows) and based on the value used for the maximum valid note duration $T_{\max}$. The metric shown is the F-score achieved by each class and the average value across all instruments.

260 ms and the 90th percentile is 0.464 ms. So, the value of 400 ms is already enough to represent the vast majority of the notes. Moreover, when the analysed note event is longer than 400 ms, the 400 ms initial window contains most of the important features for the model.

7.3 Auxiliary Input and Types of Representations

To test the importance of the auxiliary input and how its modification would affect the performance of the model, we also tested a version of the model using only the main mel spectrogram input and versions using different numbers of harmonics H in the auxiliary input (from $H = 1$ to $H = 5$). We also tested two types of input representation for the model, the Constant-Q Transform (CQT) and the mel-frequency spectrogram. The CQT was computed using 12 bins per octave and a total of 115 bins starting from $G\sharp0$ (MIDI $\#20$). The mel-frequency spectrogram was computed by a linear transformation of an STFT onto a mel-scaled frequency axis, using 256 mel-bins. The results are provided in Table 2.

Analysing the results, it is possible to say that the auxiliary input is extremely necessary for the framework. Without it, the average F-score only reaches 60.9%, while with it the performance improves up to 90.4%. Apart from piano, all other classes have a large decrease in performance when we exclude the auxiliary input from $\mathbf{X}$. We believe that the results for the piano class continue to be high not only because of the MusicNet bias towards piano, but also because some recordings of the test set are solo piano recordings, which facilitates the classification of piano notes when analysing the main input signal due to the absence of other classes. Regarding the number of harmonics used in the auxiliary input, we can see that, in general, the CQT works best with few harmonics, while the Mel-STFT prefers higher values. A possible explanation for this is the fact that it is harder to represent odd harmonics on the CQT using a log-frequency resolution of 12 bins per octave. However, more experiments are needed in order to better investigate this assumption.

7.4 Streaming of Multi-Pitch Estimations

Once we verified that our model obtains impressive performance when the original ground-truth labels are used, we tested the classifier in a more realistic environment, where no note-event labels are readily available. We estimated frame-level pitch values using two third-party MPE algorithms [8, 9]. For the algorithm in [8] we obtained an implementation from the original authors while an implementation of [9] is available via the project Omnizart[4]. We ran both algorithms on the music recordings to obtain the note events in order to construct the input to the classifier.

It is important to observe that errors in the MPE estimation will be carried over to the instrument assignment task. If a note is wrongly estimated, no ground-truth class for the instrument assignment exists, so it is hard to evaluate the results in the same way we did for the other experiments. So, in this experiment we used the transcription metrics that we explained in the last paragraph of Section 6. The results appear in Table 4. Given the limitations of each MPE method we used, we can see that our approach can successfully generate multi-instrument transcriptions.

Instr.	Onset			Onset + Offset		
	GT	[8]	[9]	GT	[8]	[9]
MPE-only	1	0.633	0.480	1	0.423	0.200
piano	0.942	0.745	0.451	0.942	0.497	0.196
violin	0.997	0.529	0.499	0.997	0.381	0.225
viola	0.775	0.366	0.308	0.775	0.227	0.116
cello	0.954	0.596	0.570	0.954	0.507	0.258
horn	0.804	0.460	0.429	0.804	0.232	0.166
bass.	0.874	0.473	0.373	0.874	0.193	0.130
clar.	0.967	0.616	0.456	0.967	0.344	0.165

Table 4. Transcription results when using Ground-Truth (GT) labels and when using two different MPE methods. In the row "MPE-only" no instrument assignment is done, we evaluate the multi-pitch estimates using the reference ground-truth notes ignoring the instrument annotations.

8. CONCLUSIONS

In this work we presented a convolutional neural network for note-level instrument assignment. We approach this problem as a classification task and proposed a framework that uses the pitch information of the note-events to guide the classification. Our approach can also successfully classify notes provided by a MPE algorithm, which permits generating multi-instrument transcriptions. Our method also shows the benefits of including different kernel shapes in the convolutional layers.

As future work we plan to investigate more deeply the interaction of our method with MPE algorithms as well as how the final estimations can be improved by including a clip-level analysis. The adoption of multi-label classification approaches is also planned.

9. ACKNOWLEDGEMENT

This work is supported by the European Union's Horizon 2020 research and innovation programme under the Marie

[4] `https://github.com/Music-and-Culture-Technology-Lab/omnizart`

Skłodowska-Curie grant agreement No. 765068 (MIP-Frontiers).

10. REFERENCES

[1] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic music transcription: An overview," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2019.

[2] Z. Duan, J. Han, and B. Pardo, "Multi-pitch streaming of harmonic sound mixtures," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 1, pp. 138–150, 2014.

[3] Y.-N. Hung and Y.-H. Yang, "Frame-level instrument recognition by timbre and pitch," in *International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, Sep 2018.

[4] Y.-N. Hung, Y.-A. Chen, and Y.-H. Yang, "Multitask learning for frame-level instrument recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton. UK, May 2019.

[5] Y. Han, J. Kim, and K. Lee, "Deep convolutional neural networks for predominant instrument recognition in polyphonic music," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 25, no. 1, pp. 208–221, 2017.

[6] S. Gururani, M. Sharma, and A. Lerch, "An attention mechanism for musical instrument recognition," in *International Society for Music Information Retrieval Conference (ISMIR)*, Delft, Netherlands, Nov 2019.

[7] J. Thickstun, Z. Harchaoui, and S. M. Kakade, "Learning Features of Music from Scratch," in *International Conference on Learning Representations (ICLR)*, New Orleans, USA, May 2017.

[8] C. Thomé and S. Ahlbäck, "Polyphonic pitch detection with convolutional recurrent neural networks," in *Music Information Retrieval Evaluation eXchange (MIREX)*, 2017.

[9] Y.-T. Wu, B. Chen, and L. Su, "Polyphonic music transcription with semantic segmentation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, May 2019.

[10] J. Pons, T. Lidy, and X. Serra, "Experimenting with musically motivated convolutional neural networks," in *International Workshop on Content-Based Multimedia Indexing (CBMI)*, Bucharest, Romania, Jun 2016.

[11] J. Pons, O. Slizovskaia, R. Gong, E. Gómez, and X. Serra, "Timbre analysis of music audio signals with convolutional neural networks," in *European Signal Processing Conference (EUSIPCO)*, Kos Island, Greece, Aug 2017.

[12] C. Lordelo, E. Benetos, S. Dixon, and S. Ahlbäck, "Investigating kernel shapes and skip connections for deep learning-based harmonic-percussive separation," in *Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, USA, Oct 2019.

[13] Y. Wu, B. Chen, and L. Su, "Multi-instrument automatic music transcription with self-attention-based instance segmentation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 28, pp. 2796–2809, 2020.

[14] A. Solanki and S. Pandey, "Music instrument recognition using deep convolutional neural networks," *International Journal of Information Technology*, pp. 1–10, Jan 2019.

[15] E. Benetos and S. Dixon, "Multiple-instrument polyphonic music transcription using a temporally constrained shift-invariant model," *Journal of the Acoustical Society of America*, vol. 133, no. 3, pp. 1727–1741, 2013.

[16] V. Arora and L. Behera, "Multiple f0 estimation and source clustering of polyphonic music audio using PLCA and HMRFs," *IEEE/ACM Transactions on Audio Speech and Language Processing (TASLP)*, vol. 23, pp. 278–287, 2015.

[17] K. Tanaka, T. Nakatsuka, R. Nishikimi, K. Yoshii, and S. Morishima, "Multi-instrument music transcription based on deep spherical clustering of spectrograms and pitchgrams," in *International Society for Music Information Retrieval Conference (ISMIR)*, Montreal, Canada, Oct 2020.

[18] R. Bittner, B. Mcfee, and J. Bello, "Multitask learning for fundamental frequency estimation in music," *arXiv e-prints*, pp. 1–13, 2018.

[19] E. Manilow, P. Seetharaman, and B. Pardo, "Simultaneous separation and transcription of mixtures with multiple polyphonic and percussive instruments," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, May 2020.

[20] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, "Deep salience representations for f0 estimation in polyphonic music," in *International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, Oct 2017.

[21] N. Takahashi and Y. Mitsufuji, "Multi-scale multi-band DenseNets for audio source separation," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, USA, Oct 2017.

[22] G. Huang, Z. Liu, L. Van Der Maaten, and K. Weinberger, "Densely connected convolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, Hawaii, Jul 2017.