TRIAD: CAPTURING HARMONICS WITH 3D CONVOLUTIONS

Miguel Perez[♯♭] **Holger Kirchhoff**[♯] **Xavier Serra**[♭]
Huawei, Munich Research Center[♯]
miguel.perez.fernandez@huawei.com
MTG, Universitat Pompeu Fabra[♭]
xavier.serra@upf.edu

ABSTRACT

Thanks to advancements in deep learning (DL), automatic music transcription (AMT) systems recently outperformed previous ones fully based on manual feature design. Many of these highly capable DL models, however, are computationally expensive. Researchers are moving towards smaller models capable of maintaining state-of-the-art (SOTA) results by embedding musical knowledge in the network architecture. Existing approaches employ convolutional blocks specifically designed to capture the harmonic structure. These approaches, however, require either large kernels or multiple kernels, with each kernel aiming to capture a different harmonic. We present TriAD, a convolutional block that achieves an unequally distanced dilation over the frequency axis. This allows our method to capture multiple harmonics with a single yet small kernel. We compare TriAD with other methods of capturing harmonics, and we observe that our approach maintains SOTA results while reducing the number of parameters required. We also conduct an ablation study showing that our proposed method effectively relies on harmonic information.

1. INTRODUCTION

When a note is played, a set of strongly related frequencies start to sound leading to a pitch sensation for the listener. These strongly related frequencies are what we call the *harmonic spectrum*, in which we distinguish two parts: the fundamental frequency (f_0) and the harmonics. The fundamental is the frequency associated with the pitch, and the harmonics are integer multiples of f_0. Different instruments reinforce different harmonics, achieving different timbres; but the underlying structure created by f_0 and its harmonics remain present.

Traditional Automatic music transcription (AMT) systems based on manual feature design employed this property to look for harmonic patterns given an observed spectrogram [1]. When DL became more popular, many researchers refrained from incorporating expert knowledge into their model architectures, but relied on generic models in combination with large amounts of task-specific training data. Even though these systems significantly outperformed traditional approaches, models utilized large numbers of parameters. [2, 3].

The number of parameters plays an important role, as more parameters can help capture the harmonic pattern better; in exchange, larger models require more computing resources as the number of operations grows. Many DL practitioners do not always have access to large GPU clusters, and might not be able to train such large models. Moreover, many portable devices such as phones have limited battery and memory, and such large models in those devices will either quickly drain their battery or be directly impossible to employ. Part of the research focused on reducing the number of models' parameters without harming the transcription's accuracy. This was achieved in many cases through the incorporation of pitch expert knowledge within the architecture neural network (NN) [4–9].

The main challenge resides in the unequal distances between harmonics in the spectrum, so previous approaches employ either large kernels or several ones running in parallel. This paper introduces a ***tri***dimensional kernel h***a***rmonically ***d***ilated (TriAD), a neural block that captures music intervals and is capable of observing multiple harmonics while using a single yet small kernel.

The rest of the paper is divided into the following sections: Section 2 gives more details about prior work capturing harmonics from the spectrum. Section 3 describes our method, including the processing of the signal and the design of the kernels. The experimental setting is described in Section 4. We present the results for these experiments as well as an ablation study in Section 5. Finally, Section 6 contains our conclusions for this paper and future work.

2. RELATED WORK

As mentioned in Section 1, harmonics played an important role in the first AMT systems. For example, [1] creates a dictionary of sets of expected harmonics for each fundamental. These ideal patterns were then matched to the spectrograms used as input for the system using the non-negative least squares (NNLS) algorithm. The result is an estimation of fundamental frequencies that along with their respective harmonics, would resemble the input's spectrogram.

For AMT systems using DL, prior work has incorporated domain-specific knowledge in two ways: 1. by choosing a custom input representation that allows the model to detect harmonic structures [4, 10, 11]; 2. by employing specific network architectures to search for pat-

terns in a given feature map obtained at any point of the network [6–8, 12]. Within the first category, one of the most popular approaches is the harmonic constant Q transform (HCQT) [4], a feature that extends the constant Q transform (CQT) [13]. The standard CQT returns a log-frequency representation of the spectrum, where the n^{th} bin is associated with the frequency $f_n = f_{min} \cdot 2^{n/p}$ where f_{min} is the minimum frequency to be considered, and p is the number of bins per octave. The magnitude of CQT spectrogram is a representation containing a single channel, F_{bins} frequency bins, for T frames; its shape is $[1, F_{bins}, T]$. The HCQT extends the CQT the channel dimension, where now H harmonics are aligned, resulting in a tensor with dimensions $[H, F_{bins}, T]$. This extension is done by stacking a number of H CQTs through the channel dimension. Each one of these H CQTs is a regular one whose f_{min} has been scaled by a harmonic factor h: $f_n = h \cdot f_{min}$; the CQTs with $h = 1$ will refer to the fundamental, $h = 2$ will refer to the first harmonic, $h = 3$ to the third harmonic, etc. up to H different values. Similarly, sub-harmonics can be added by making $h = 0.5, 0.25$, etc. In a nutshell, the HCQT facilitates information about the fundamentals directly at the network's input.

As mentioned, other works incorporated the harmonic knowledge within the architecture of NNs, e.g. [6] extended the idea of frequency-shifted representations, for the internal feature maps obtained inside NNs. The authors named this method multiple rates dilated harmonic causal convolution (MRDC-Conv). Let $\mathcal{X}$ denote a feature map, with shape $[C_{in}, F_{bins}, T]$ at an arbitrary point of the network. The number of channels for that map is C_{in}. In a CQT spectrum, the distance d_n between the fundamental frequency and the n^{th} harmonic is given by:

$$d_n = \text{round}(p \cdot \log_2(n)) \tag{1}$$

Where p is a parameter that determines the number of bins per octave in the CQT spectra. To capture k harmonics with MRDC-Conv, the feature map $\mathcal{X}$ is convolved with k different kernels in parallel, resulting in k outputs. Each of the outputs is shifted following the harmonic factors given by Equation 1. E.g. to capture the first three harmonics, three different kernels are required, thus, producing three different outputs. In the case of $p = 12$ and following Equation 1, the shifts associated with the $2^{nd}, 3^{rd}$ and $4^t h$ harmonics are 12, 19, and 24. The sum across the k outputs is taken, leading to a single final output of shape $[C_{out}, F_{bins}, T]$, where C_{out} is the number of output channels. This method is illustrated in Figure 1a. MRDC-Conv achieves a convolution able to observe the input at the precise position of the harmonics; its drawback is that for each of the harmonics, a different kernel is needed, thus requiring a different feature map stored in memory for each of the k harmonics before they can be aggregated.

Some other authors embedded harmonic knowledge within the convolutional kernels rather than in the manipulation of their inputs/outputs. In [12] the authors use sparse convolutions so that only relevant parts of the spectrum are considered. Sparse convolutions allow the kernels to "ignore" certain parts of the input, so they do not contribute either to the output or to backpropagation during training [14]. According to [15], the harmonics are *positive* indicators that a certain pitch is present, but some frequencies indicate that the pitch might not be present at all. The latter are called *negative* indicators. The sparse convolutions from [12] are used in such a way that only *positive* and *negative* indicators defined in [15] are taken into account. Sparse convolutions require nonetheless using large kernels to cover relevant parts of the spectrum, i.e. [12] resulted in around 650k parameters exclusively for harmonic processing, accounting for the major portion of the model's parameters.

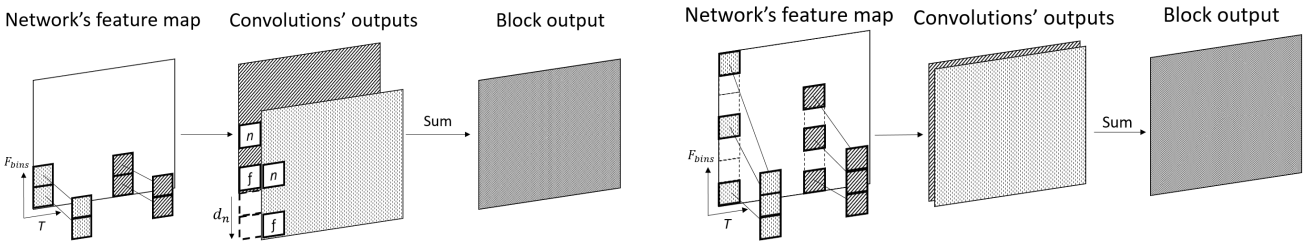
Harmonics	Music Interval	pitc class distance
$2, 4, 8, 16$	octave	$b \cdot 12$
17	minor second	$b \cdot 1$
$9, 18$	major second	$b \cdot 2$
19	minor third	$b \cdot 3$
$5, 10, 20$	major third	$b \cdot 4$
21	perfect fourth	$b \cdot 5$
$11, 22$	augmented fourth	$b \cdot 6$
$3, 6, 12, 24$	perfect fifth	$b \cdot 7$
25	minor sixth	$b \cdot 8$
27	major sixth	$b \cdot 9$
$7, 14, 28$	minor seventh	$b \cdot 10$
$15, 30$	major seventh	$b \cdot 11$

Table 1: The harmonics of the first 3 octaves, and their associated music intervals. The rightmost column indicates the distance in bins associated with each interval, where b is the number of bins per semitone.

In [8], dilated convolutions are used to capture the harmonics from the spectrum, with a method named harmonic dilated convolution (HD-Conv). Dilated convolutions are a special kind of convolution, where the kernels' inputs are spaced by a fixed amount. An example of dilated convolutions can be seen in Figure 1b. By controlling the dilation size, the authors space the kernels' inputs, so each kernel obtains a specific harmonic. The outputs of the different kernels are aggregated by summing across the kernels' outputs as shown in Figure 1b. The size of the dilations is given by Equation (1). E.g. for $p = 12$, the second harmonic is separated from the fundamental by $d_2 = 12$ bins, the third one by $d_3 = 19$; to capture both the second and the third harmonic, we would need to create two convolutional kernels with a dilation size of 12 and 19 at the frequency dimension. This method has the same drawback as MRDC-Conv, as different harmonics also require a different kernel.

3. OUR METHOD

Similarly to [8], our method uses dilated convolutions to capture the harmonics of the spectrum. As mentioned before, a constant dilation can not capture multiple harmonics given the logarithmic nature of these. If it was possible to use different dilations for the same kernel, this problem



(a) MRDC-Conv

(b) HD-Conv [8]

Figure 1: Figure (a) An example of MRDC-Conv [6]. Two kernels are applied to the same input. The fundamental f is separated from the harmonic n by d_n bins. One output gets shifted by d_n, and so f and n get aligned. Figure (b) An example of HD-Conv [8], with two kernels applied to the same input, each one with a different dilation (3, and 2 respectively).

would have been already solved, but currently, DL frameworks support only dilations with constant spacing. Our method is able to partially overcome this technical limitation and achieve a convolution at the frequency axis with different dilation rates; thanks to this, our proposed method captures multiple harmonics by just using a single kernel.

We named our method *TriAD*, and it involves a series of steps. The first step is to split the frequency dimension into two new ones, each representing different octaves and pitch classes. We call this representation the *pitch/octave* spectrogram. Next, we create the kernels for our method. Previous works used kernels spanning 2 dimensions: frequency and time; our method's kernels however span 3 dimensions: octave, pitch class, and time. An arbitrary number of m different kernels can be created, each one capturing a different music interval. The m kernels are convolved with the previously described pitch/octave spectrogram, resulting in m different outputs. Finally, these outputs are aggregated by taking the sum across them. The consecutive steps are illustrated in Figure 2.

Subsection 3.1 details the procedure followed to convert a log-frequency spectrogram onto a pitch/octave spectrogram. Subsection 3.2 explains how our convolutional kernels are created and the difference they have with the method described in [8]. At the end of that subsection, we describe a special kind of padding used in our technique, the octave-circular padding.

3.1 The pitch/octave spectrogram

Let $\mathcal{X}^{C_{in} \times F_{bins} \times T}$ be a feature map, with F_{bins} logarithmically spaced frequency bins, T frames, and C_{in} channels. Our goal is to separate octave and pitch class information. We split the F_{bins} bins into two dimensions representing the octave (o) and pitch class (p) information. The number of pitch classes is simply the number of bins per octave used, and the number of octaves can be obtained by $o = \frac{F_{bins}}{p}$. Note that o must be an integer, and so when this condition is not met, we pad the upper part of $\mathcal{X}$'s frequency dimension with the minimum amount of zeros that satisfies the condition. The result is the pitch/octave spectrogram $\mathcal{Y}^{C_{in} \times o \times p \times T}$, a view of $\mathcal{X}$ where F_{bins} has been separated into its octave and pitch class information.

3.2 The harmonic convolutions

Our aim is to compare two pitch classes across multiple octaves to capture harmonically related information. As shown in Table 1, harmonics and music intervals are closely related. Comparing two pitch classes separated by a certain interval at multiple octaves simultaneously will effectively obtain the harmonics associated with that music interval.

As previously mentioned, our kernels have 3 dimensions: $K^{k_o \times k_p \times k_t}$, related to the octaves (k_o), pitch classes (k_p), and frames (k_t) of the pitch/octave spectrogram; this means that our method uses 3D convolutions [1]. By changing the convolution dilation at the pitch class dimension we control which interval we capture, and consequently its associated harmonics. Since our goal is to compare the same two pitch classes, our method has a fixed $k_p = 2$, but the sizes of k_o and k_t can be varied, spanning many octaves and timesteps. The effect of dilation exclusively on pitch classes is what achieves the aforementioned non-constant dilation at the frequency dimension. E.g. Let $p = 12$ and a kernel K with $k_o = 3$ and a perfect fifth dilation at the pitch class dimension, in a certain position, this kernel would see $C_1, G_1, C_2, G_2, C_3, G_3$ simultaneously. The distance from each C to the next G is 7 bins, but the distance from each G to the next C is 5 bins. Our method is to the best of our knowledge, the only one capable of achieving that effect in dilation. In the same scenario using linear dilations [8], a kernel with the same size and dilation of a perfect fifth would see instead $C_1, G_1, D_2, A_2, E_3, B_3$. Using our method, a single kernel with $k_o = 3$ and a dilation of perfect fifths at the k_p dimension capture 5 of the first 7 harmonics (see Table 1).

As can be observed in Figure 2, the inputs and outputs of the convolutions have the same size, which is achieved by padding the pitch/octave spectrogram. The values used to pad follow the values of the continuous log-frequency spectrogram. E.g. given $p = 12$, to pad above B_1, we use the values of the bins C_2,$C\sharp_2, etc.$ In contrast, values above the highest octave of the pitch/octave spectrogram will be padded with zeros. We call this method *circular-octave padding*.

[1] When $k_t = 1$, our method can be implemented with 2D convolutions by stacking frames across the batch dimension. 3D is just the general case for an arbitrary k_t

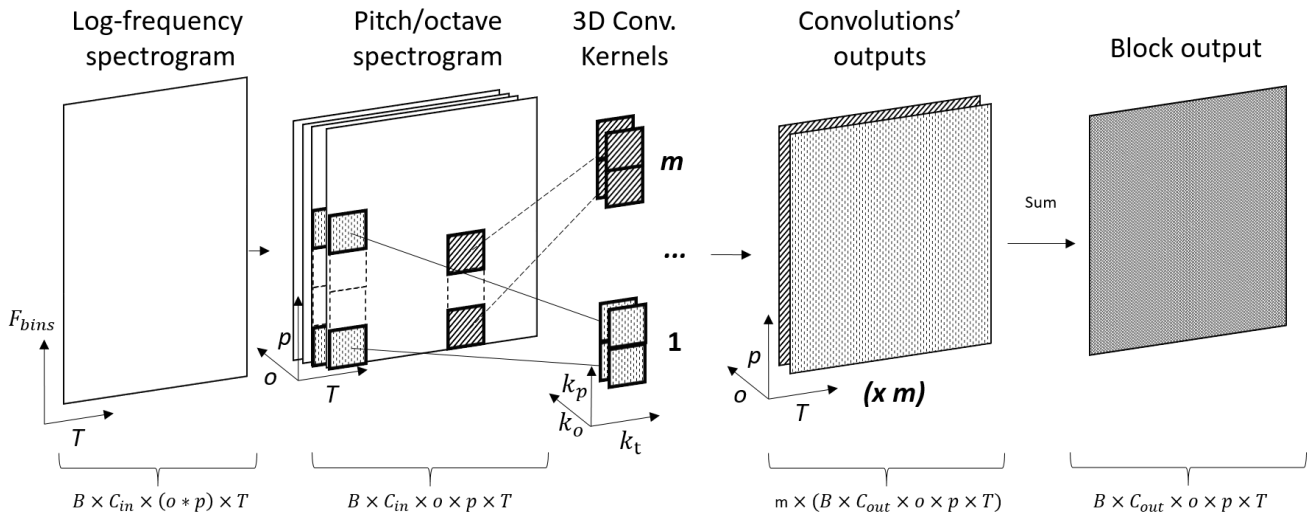


Figure 2: An overview of TriAD. The channel dimension has been omitted in the image. The first stage converts a log-frequency spectrogram onto a pitch/octave one. We apply m of our harmonically motivated kernels to the pitch/octave spectrogram. Each kernel captures different harmonics, depending on the dilation at the p dimension. The kernels' outputs are aggregated by summing the m outputs. B stands for the batch dimension.

4. EXPERIMENTS

We test the performance of our method on AMT for the subtask of piano transcription. Our method is compared with other SOTA approaches of capturing the harmonic spectrum within the architecture itself; concretely, we used the harmonic blocks MRDC-Conv [6], and HD-Conv [8]. We do not include input manipulations such as the HCQT, since these are input manipulations rather than network-internal musically motivated convolutional operations, and a fair comparison is not straightforward.

4.1 Datasets

We used two datasets in our experiments: *MIDI and audio edited for synchronous track and organization* (MAESTRO) [16], and *MIDI aligned piano sounds* (MAPS) [17]. MAESTRO contains about 200 hours of audio for complex piano performances precisely aligned to note labels. Some compositions appear multiple times, each played by a different interpreter. In the paper where MAESTRO is presented, an official train/validation/test configuration was also proposed so that compositions played by different interpreters are in the same split group. We use the latest version of this dataset, version 3, in our experiments. MAPS is another popular dataset used in piano transcription. In contrast to MAESTRO that contains only complete piano pieces, this dataset also contains isolated notes and chords.

Following the practice used in previous works [7,8,16], we use the train and validation splits from MAESTRO to train our NNs, and the test sets of MAESTRO and MAPS for testing the trained models. Chunks of audio of 20 seconds and a sample rate of 16.000Hz were used and transformed into a CQT spectrogram, with 352 bins, $f_{min} = 32.070Hz$, and a resolution of 4 bins per semitone. A hop size of 320 samples is employed, resulting in a time resolution of 20 milliseconds.

4.2 The model

We use the HPPNet-base model from [8] for our experiments. This model consists of a backbone and 4 different heads; each head is in charge respectively of predicting which notes are present in each frame, its velocity and whether there is an onset or offset happening. Figure 3 shows an overview of the network. The backbone consists of multiple convolutional layers, and it is divided into three main sections. The first section consists of 3 blocks with 2D convolutions, whose kernels are squarely shaped (7×7) and perform initial processing of the CQT spectrogram. The second section is in charge of doing the backbone's harmonic processing; this is where either HD-Conv, MRDC-Conv, or TriAD will be placed. The last block consists of 5 2D convolutional layers with filter shape (1×5), spanning across the time dimension [2].

The output of the backbone is then used as input for the four heads. Each head consists of a bidirectional long short-term memory (LSTM) [18] and a dense layer. LSTMs model sequential data, which are the features associated with each output bin in this case. The dense layer takes the features outputted by the LSTM and produces a single value for each of the 88 notes of a piano. Details about the design choices of HPPNet can be found in [8].

We run our experiments by comparing the model's performance when the backbone's harmonic processing is done either by our method (TriAD), MRDC-Conv [6], or HD-Conv [8]. We use those methods as employed in their respective papers: 12 kernels of shape (1×1) in the case of [6], and 8 kernels with shape (3×1) in the case of [8]. For our method, we use just two kernels, one dilated for perfect fifths, and another one for major thirds; these are

[2] The third block differs from the original paper description; following their description, that block of the backbone alone has 983.040 parameters, whereas the paper specifies that the backbone contains 421K parameters. We used the network as implemented in the official repo, which matches the number of parameters and replicates their reported results

the intervals with the most associated harmonics. Our kernels span 3 octaves ($k_o = 3$) and a single frame ($k_t = 1$). The code for MRDC-Conv and HD-Conv can be found in their official repositories [3] [4]. We do not train a version of the model with a "harmonically agnostic" block, as [8] already shows in an ablation study that the model's performance drops significantly in that case.

As optimizer, ADAM [19] with a learning rate of $6 \cdot 10^{-3}$ was used. We trained all the models for 200.000 steps, where each step consists of a batch size of 4 chunks of audio. The evaluation was done on MAESTRO's evaluation dataset every 500 steps, to check for possible cases of overfitting. The models were trained 3 times, each one with a random weight initialization. All the harmonic blocks take a similar time to train, around 24h to complete in a V100 GPU.

The employed loss is the same one as in HPPNet's paper [8], a combination of individual losses for the frame, onset, offset, and velocity heads. Weighted binary cross entropy (see Equation 2) was used as loss for the frame, onset and offset heads. This loss is used since there are few positive onset labels, yet predicting onsets is necessary to distinguish consecutive notes. The parameter w controls the relevance of positive labels in the loss and is chosen as $w = 1$ for offsets and frames, and $w = 2$ for onsets. The loss for the velocity head is the mean squared error between the expected and estimated velocities of each individual note.

$$l_{bce}(y, \hat{y}) = -wy \cdot \log(\hat{y}) - (1 - y) \cdot \log(1 - \hat{y}) \quad (2)$$

5. RESULTS

The metrics reported follow the convention described in [20]. These metrics report different aspects of the transcription. The *frame* metric operates at the frame level, while the other three operate at the *note* level. Within the note level, three different metrics exist, considering offsets and/or velocity. This is due to the partially subjective nature of this task. The onset (referred to as the moment when a certain note starts to sound) is not very subjective given the sharp attack of the piano [21]. In contrast, offset (the moment when a certain note stops sounding) and velocity are less objective aspects of the transcription. An estimate of a note is assumed to be correct if its onset is within ± 50ms of the reference, and its pitch is correct. When contemplating offsets, in addition to the previous requisites, the estimation's offset should also be within a certain range; this range is either ± 50ms or 20% of the reference note's duration, whatever is larger.

Velocity estimation is more intricate, as depending on the microphone position a note played with a certain velocity can sound louder or quieter. We use the procedure described in [2], which involves rescaling velocities and using linear regression to account for the aforementioned difference in loudness. All the metrics were calculated using *mir_eval* [22].

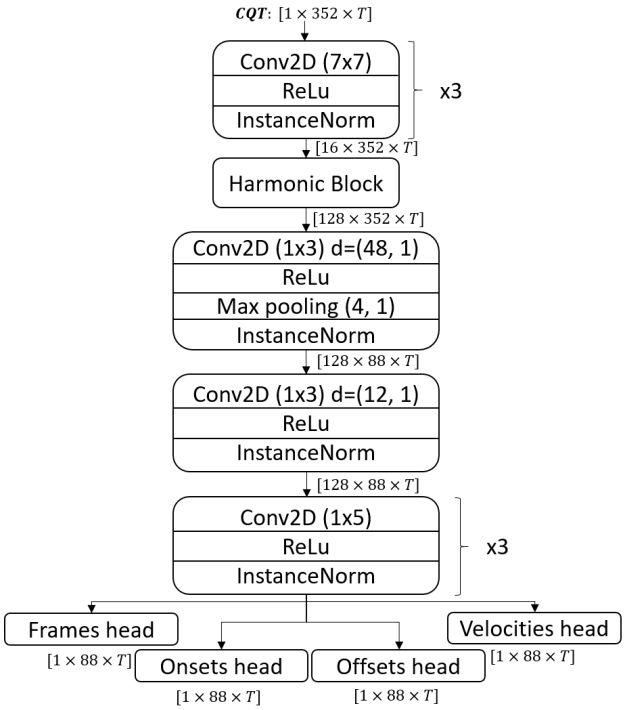


Figure 3: A diagram of HPPNet. The brackets' numbers represent the sizes of the channel, frequency, and frame dimensions. Letter d indicates the dilation rate.

The scores for Section 4 experiments are in Table 2. We also report the results of some larger models of the SOTA as reference. Onsets & Frames [2] is among the most well-known DL models for piano transcription. Semi-CRFs [3] is a method designed to improve the predictions made about the offsets. These are large and capable models, but the ones using harmonic knowledge also manage to achieve similar results with notably fewer parameters. Both TriAD and HD-Conv blocks achieve similar results, in pair with large models. The MRDC-Conv block uses fewer parameters than HD-Conv, but in exchange drops in performance. Noticeably, the model using TriAD has the same number of parameters as the one MRDC-Conv, yet it does not drop in performance.

5.1 Kernel dilation relevance

In music theory, some intervals are more important than others. Equally, some music intervals have more harmonics associated with them than others, as shown in Table 1. It could be expected, that using a kernel dilated with a highly relevant interval yields better results than a kernel associated a with less relevant interval. We tested whether this assumption held or not in our method; instead of using multiple kernels as previously described, our block consists of a single kernel for these experiments. We used 2 relevant intervals (perfect fifth, major third), and 2 lesser relevant intervals (minor second, major seventh) to test the aforementioned assumption. These kernels span 3 octaves ($k_o = 3$) and a single frame ($k_t = 1$), as in the previous ex-

[3] https://github.com/WX-Wei/HarmoF0
[4] https://github.com/WX-Wei/HPPNet

Model	# Parameters	FRAME F1	NOTE F1	NOTE W/OFFSET F1	NOTE W/OFFSET & VEL. F1
		MAESTRO			
Onsets & Frames [2]*	26M	89.68%	95.22%	79.44%	78.85%
Semi-CRFs [3]	9M	90.75%	96.11%	**88.42%**	**87.44%**
HPPNet + HD-Conv	820K	**91.62%**(±.02)	96.14%(±.01)	82.91%(±.02)	80.91%(±.02)
HPPNet + MRDC-Conv	780K	78.69%(±.01)	84.71%(±.01)	58.77%(±.01)	52.15%(±.03)
HPPNet + TriAD (ours)	780K	91.50%(±.02)	**96.16%**(±.01)	82.62%(±.02)	80.76%(±.01)
		MAPS			
HPPNet + HD-Conv	820K	**72.45%**(±.02)	**86.09%**(±.01)	**42.77%**(±.02)	40.11%(±.02)
HPPNet + MRDC-Conv	780K	63.25%(±.01)	73.87%(±.02)	32.68%(±.02)	32.68%(±.01)
HPPNet + TriAD (ours)	780K	72.39%(±.03)	85.06%(±.02)	42.41%(±.02)	**40.17%**(±.02)

Table 2: Results for the experiments described in Section 4. In our experiments, each model was trained three different times. The metrics here reported are the average across these runs and in parenthesis the variance. * Results from [8].

Model	Major third		Perfect fifth		Minor second		Major seventh	
	MAESTRO	MAPS	MAESTRO	MAPS	MAESTRO	MAPS	MAESTRO	MAPS
HPPNet + TriAD	**90.14%**(±.02)	**71.58%**(±.01)	**90.23%**(±.02)	**71.98%**(±.01)	83.16%(±.01)	**68.53%**(±.01)	83.36%(±.01)	**69.19%**(±.02)
HPPNet + HD-Conv	84.89%(±.01)	69.96%(±.02)	85.98%(±.02)	70.50%(±.03)	**84.23%**(±.03)	67.86%(±.03)	**84.79%**(±.01)	68.69%(±.02)

Table 3: F1 framewise results for the single kernel experiments described at section 5.1. Our method obtains worse results if a "less relevant" music interval is chosen. HD-Conv achieves more similar results regardless of the dilation, with just a small improvement for the case of the perfect fifth (where it employs two kernels).

periment. We also used the method with constant dilations i.e. HD-Conv from [8], equally using single kernels except for the case of the perfect fifth. There are two harmonics associated with the perfect fifth within the first 3 octaves, so we employ two rather than a single kernel. The constant dilations capture in this case major third: $5th$ harmonic; perfect fifths, $3rd$ and $6th$ harmonics; minor second, $17th$ harmonic; and major seventh $30th$ harmonic. We noticed that after 50.000 steps, the speed at which the loss diminished slowed down sensibly, and therefore, we reduced the number of training steps for this experiment and trained for 70.000 steps in each run.

The results can be seen in Table 3. HD-Conv [8] obtains slightly better results for the perfect fifth kernels, but similar results for other cases. Our method (TriAD) has a distinguishable performance gap depending on the interval. Results are worse for minor second and major seventh intervals, compared to the cases of the major third and the perfect fifth. Moreover, in those two cases, our method achieves notably better results than HD-Conv.

6. CONCLUSIONS

In this paper, we presented TriAD, a novel convolutional block for NNs capable of capturing the harmonics related to music intervals. To obtain such information, we separate octave and pitch class dimensions from log-frequency spectrograms and create convolutional kernels specifically designed to process this disentangled representation. We tested and compared our method with other ones designed to capture harmonic information, in the task of piano-AMT. We also compared how our model performed when only a single kernel was employed. To the best of our knowledge, our method is the only one capable of achieving dilated convolutions which are not "equally spaced" along the frequency axis, allowing our model to capture multiple harmonics using a small kernel. To achieve this effect, other approaches require applying different convolutional layers to the same input [6, 8] or using large kernels [12].

Our method is still capable of reaching the performance achieved by other harmonic blocks while making use of fewer parameters, showing the effectiveness of our approach. Furthermore, the results from the experiment described in Subsection 5.1 show that our method's performance highly depends on the dilation choice, thus hinting that our method is indeed using the harmonics to determine which pitches are present. Moreover, with an appropriate dilation choice our model outperforms other methods also using a single kernel.

Harmonic series are relevant for other tasks beyond AMT, for example, instrument recognition. Some works have found that the harmonics and their respective amplitudes are crucial to correctly classifying instruments [23, 24]. Our method could be employed to capture the amplitude of different harmonics and learn specific patterns for each instrument. In future work, we will use "harmonically designed" networks in other AMT related tasks. Recent advances in AMT such MT3 [25] demonstrate that with the current DL techniques is possible to transcribe an arbitrary number of instruments from a piece of music audio instead of just piano as shown here. Since the harmonics are relevant for instrument recognition, we hypothesize using harmonic blocks such as the ones presented here, the accuracy with which notes are assigned to each instrument in systems like MT3 could improve. We release code for reproducibility experimentation [5] .

[5] https://github.com/migperfer/TriAD-ISMIR2023

7. REFERENCES

[1] M. Mauch and S. Dixon, "Approximate note transcription for the improvedidentification of difficult chords," in *Proceedings of the 11th International Society for Music Information Retrieval Conference*, 2010.

[2] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, "Onsets and Frames: Dual-Objective Piano Transcription," in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, 2018.

[3] Y. Yan, F. Cwitkowitz, and Z. Duan, "Skipping the frame-level: Event-based piano transcription with neural semi-crfs," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34, 2021.

[4] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, "Deep Salience representations for F0 estimation in polyphonic music," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, 2017.

[5] Jiří Balhar and Jan Hajič jr., "Melody extraction using a harmonic convolutional neural network," MIREX Melody Extraction Report, Tech. Rep., 2019.

[6] W. Wei, P. Li, Y. Yu, and W. Li, "HarmoF0: Logarithmic Scale Dilated Convolution for Pitch Estimation," in *2022 IEEE International Conference on Multimedia and Expo (ICME)*, Jul. 2022.

[7] X. Wang, L. Liu, and Q. Shi, "Enhancing Piano Transcription by Dilated Convolution," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec. 2020.

[8] W. Wei, P. Li, Y. Yu, and W. Li, "HPPNet: Modeling the Harmonic Structure and Pitch Invariance in Piano Transcription," in *Proceedings of the 23th International Society for Music Information Retrieval Conference*, 2022.

[9] R. M. Bittner, J. J. Bosch, D. Rubinstein, G. Meseguer-Brocal, and S. Ewert, "A Lightweight Instrument-Agnostic Model for Polyphonic Note Transcription and Multipitch Estimation," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2022.

[10] V. Lostanlen and C. Carmine-Emanuele, "Deep convolutional networks on the pitch spiral for musical instrument recognition," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, 2017.

[11] J.-F. Ducher and P. Esling, "Folded cqt rcnn for real-time recognition of instrument playing techniques," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, 2019.

[12] X. Wang, L. Liu, and Q. Shi, "Harmonic Structure-Based Neural Network Model for Music Pitch Detection," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec. 2020.

[13] J. C. Brown, "Calculation of a constant *Q* spectral transform," *The Journal of the Acoustical Society of America*, vol. 89, no. 1, Jan. 1991.

[14] B. Graham, M. Engelcke, and L. v. d. Maaten, "3d semantic segmentation with submanifold sparse convolutional networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[15] A. Elowsson, "Polyphonic pitch tracking with deep layered learning," *The Journal of the Acoustical Society of America*, vol. 148, no. 1, 2020. [Online]. Available: https://doi.org/10.1121/10.0001468

[16] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, Cheng-Zhi, A. Huang, S. Dieleman, E. Erich, J. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the maestro dataset," in *Proceedings of the 7th International Conference on Learning Representations*, 2019.

[17] V. Emiya, R. Badeau, and B. David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, 2010.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, 1997.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[20] J. Salamon, "Melody extraction from polyphonic music signals," Ph.D. dissertation, Universitat Pompeu Fabra, Barcelona, Spain, 2013.

[21] A. Ycart, L. Liu, E. Benetos, and M. T. Pearce, "Investigating the perceptual validity of evaluation metrics for automatic piano music transcription," *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, 2020.

[22] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P.W., "mir_eval: A transparent implementation of common mir metrics," in *Proceedings of the 15th International Society for Music Information Retrieval Conference*, 2014.

[23] Y. Mo, "Music timbre extracted from audio signal features," *Mobile Information Systems*, vol. 2022, Jun 2022. [Online]. Available: https://doi.org/10.1155/2022/1349935

[24] A. Livshin and X. Rodet, “The significance of the non-harmonic "noise" versus the harmonic series for musical instrument recognition,” in *Proceedings of the 7th International Society for Music Information Retrieval Conference*, 2006.

[25] J. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. Engel, “MT3: Multi-task multitrack music transcription,” in *International Conference on Learning Representations (ICLR)*, 2022.