Training a convolutional neural network for note onset detection on the clarinet

Tairone N. Magalhães[1] **, Mauricio A. Loureiro**[1]

[1]CEGeME - Center for Research on Musical Gesture and Expression - UFMG
Av. Antônio Carlos, 6627 – 31270-010, Belo Horizonte, MG

tairone@ufmg.br, mauriciomloureiro@ufmg.br

Abstract. *Although computational models for note onset detection have improved drastically in the last decade, mainly due to the advances brought by the field of Deep Learning, such models have not been perfected yet. When dealing with specific data, like clarinet recordings, those models still produce a significant number of false positives and negatives. In this paper, we evaluate pre-trained onset detection models from the library madmom on a dataset composed of solo clarinet recordings, in particular, to investigate their performance on this kind of data. Moreover, we use the clarinet dataset to train the same neural network (CNN) employed in one of those models, to investigate whether training the model on this specific data leads to an improvement when dealing with clarinet recordings. The results obtained from the model trained strictly on clarinet data are considerably better than those from models trained on generic data.*

1. Introduction

Note onset detection consists in finding, in audio signals, the instants at which musical notes start. The automatic detection of note onsets is a difficult task. Although some of the onset detection methods/models currently available produce excellent results in recordings of instruments with prominent note attack, their performance can drop significantly when dealing with notes with very soft attack, like woodwind instruments, for example. Clarinet recordings seem to be particularly difficult, compared to other instruments.

Many studies on automatic onset detection have been published since the beginning of the 2000s. In the beginning, the onset detection methods were mostly based on DSP techniques [1, 2, 3, 4, 5]. Recently, the majority of the studies have been employing machine learning, producing models that achieve better results on the task. However, most of these models are trained on mixed datasets, frequently containing recordings of several different instruments and musical genres. Since machine learning algorithms learn to perform tasks based on patterns that are extracted from input data, the quality of the training dataset has a pivotal impact on the quality of the final model. Having a good dataset that captures the characteristics of "real world" data is key to obtaining a model with good performance. Although it is important to avoid biases in the dataset that may negatively impact the model's generalization capacity, in some circumstances, a model might benefit from specialized data that is biased towards the expected final purpose of the model.

This led us to question whether training a machine learning model on a dataset that contains only clarinet recordings would produce a model that performed better on clarinet onsets. In this paper we test and evaluate three onset detection models from the library *madmom* [6] on clarinet recordings, specifically. Two of those models are based on neural networks, a convolutional neural network (CNN) and a recurrent neural network (RNN), and were previously trained by the author of the library on a generic dataset. The other method uses strictly DSP techniques and is based on spectral flux. We also test and evaluate a model trained by ourselves using a clarinet dataset, specifically. This model uses the same CNN architecture from madmom's model. Thus, we can check whether this specific training data leads to a better note onset detection model for the instrument.

2. Methodology

To obtain robust segmentation models using machine learning, it is necessary to provide a dataset with labeled target values. A labeled dataset also allows us to evaluate the models and compare the results they produce. Therefore, in this section we first introduce the two datasets used in our experiments. Then we detail the onset detection models employed in our experiments, followed by an explanation of the evaluation criteria we adopted.

2.1. Datasets

We created two different datasets for this experiment, both containing strictly solo clarinet recordings. To annotate the note onsets, we used a specialized web application for audio annotation developed by ourselves, called *Audio Segment Annotator*[1]. This tool seeks to provide a fine temporal resolution of 2 miliseconds, in a user interface with a steep learning curve and straightforward controls. It provides visualizations of the audio waveform and its spectrogram, synchronized with respect to the time axis.

2.1.1. Clari-onsets-50

Most datasets available made available by researchers for training note onset detection models do not focus on monophonic recordings or do not contain a significant amount of clarinet onsets. Therefore, we decided to create a large dataset composed exclusively of clarinet recordings. This enables us to train, validate and test a specialized onset detection model for clarinet. We refer to this dataset as

[1]This tool was developed for internal use in our laboratory, and has not been published yet.

absolute error (ms)	count	percentage	cumulative percentage
0 to 4 ms	541	85.9	85.9
4 to 8 ms	67	10.6	96.5
8 to 12 ms	16	2.5	99.0
12 to 16 ms	4	0.6	99.7
16 to 20 ms	1	0.2	99.8
$>$ 20 ms	1	0.2	100.0

Table 1: Absolute error for the manual annotations obtained for the dataset *clari-onsets-3*.

clari-onsets-50. It is composed by 50 excerpts of solo clarinet recordings, totaling 23 minutes and 6 seconds of audio, with a mean duration of 27.7 seconds. Half of those recordings were made in our laboratories, while the other half was obtained from comercial recordings from several different albums and clarinetists. They comprised a few different musical genres, including classical, jazz, and contemporary pieces. The recordings are very diverse with respect to level of reverberation, distance of the microphone, background noise, etc. This was intentional, since a diverse dataset, with respect to those characteristics, tend to lead to a model that is able to generalize better when the characteristics of the audio change. The excerpts were all annotated by a single person, without repetition, over several weeks. A total of 3551 note onsets were obtained for the 50 excepts.

2.1.2. Clari-onsets-3

The dataset *clari-onsets-3* contains only three recordings, with a total of 126 note onsets. The recordings were all made in the laboratory, in a room with very little reverberation. The excerpts were chosen seeking to achieve a diversity of musical material, with notes in different registers of the instrument, melodies containing both short and long intervals, notes with soft and sharp attacks, and different dynamics. For this particular dataset, we repeated the onset annotations five times (all by the same person), seeking to obtain: (1) a reliable estimation of the note onset time, based on multiple measurements; and (2) an estimation of the measurement error for the task. The annotation errors were estimated by calculating the time difference betweeh each annotated onset (from the different sessions) and the median, and are shown in table 1. This dataset is rather small, and was not used for training any neural network, but we used it as a test dataset in our experiments.

2.2. Onset Detection Models

2.2.1. CNN

This method was proposed by Schlüter and Böck, and is based on a convolutional neural network applied to mel-scaled spectrograms [7]. The advantage of this representation is that it uses a logarithmic scale for the frequency bins instead of a linear one. This logarithmic scale models better the human perception of pitch, allowing the spectrogram to have a better perceptual pitch resolution using a relatively small number of bins, thus reducing the model's memory demand.

The input of the model consists of three stacked 80-band mel spectrograms with logarithmically scaled magnitudes, each calculated using a different window length: 1024 (23 ms), 2048 (46 ms), and 4096 (93 ms) samples. They are calculated using a hop length of 441 samples, which corresponds to a frame resolution of 10 ms. The input of the network consists of a group of 15 contiguous spectral frames centered on the frame to be classified. So each example processed by the network corresponds to a context of 150 ms. The onset annotations (which consist of time points) are converted to target values, which are defined for each input example, using the criteria described below.

- If there is an onset annotation within the 10 ms time window corresponding to the central frame for a given input, the target value is assigned to 1 (onset).
- Since the annotation error might be a bit larger than this window, each frame immediately before or after a frame that was set to 1 in the step above is also set to 1, giving a margin for the network to learn from annotations that are not precise (fuzziness). These samples are also weighted with a factor of 0.25 during training.
- The targets for any other frames are defined to 0 (no onset).

Basically, the architecture of the network consists of two convolutional layers (2D convolutions) followed by an intermediary fully-connected layer, and the output layer, which is also fully-connected. The network architecture is composed by the following layers:

- input: 15 frames x 80 bands;
- 2D convolutional layer: filters of 7 frames x 3 bands, computing 10 feature maps;
- max-pooling over 3 bands with no overlap;
- 2D convolutional layer: filters of 3 x 3, computing 20 feature maps;
- max-pooling over 3 bands with no overlap;
- fully connected layer with 256 units;
- fully connected layer with a single unit (output).

All the layers use rectified linear unit activation functions. Dropout is applied to the inputs of the fully connected layers for regularization. The network is optimized using gradient descent, minimizing binary cross-entropy loss. It is trained for 100 epochs, using a starting learning rate of 1.0, which is multiplyed by 0.995 at the end of each epoch. The initial momentum is 0.45, and it is linearly increased to 0.9, between epochs 10 and 20. The mini-batches used for each training step consist in 256 examples.

The network produces output values between 0 and 1, representing the frame-wise probability of onset along the audio signal. To obtain the onset points, first, this time series is convoluted with a Hamming window of 5 frames, which smooths it a bit. Then a peak-picking

method is used to extract the local peaks higher than a given threshold value. This threshold is chosen by varying its value and picking the one that produces the highest F-score. The instants of the peaks correspond to the onsets detected by the model.

2.2.2. RNN

The model proposed by Eyben et al. [8] uses as input a stacked array of features consisting in: (1) a mel spectrogram calculated using a window length of 1024 samples (23 ms); a mel spectrogram computed using a window length of 2048 samples (46 ms); (3) the positive first-order difference between two successive mel spectrogram frames at the 1024 samples window length; (4) the positive first-order difference at the 2048 samples window length. The positive first-order difference is calculated by applying a half-wave rectifier function $H(x) = \frac{x+|x|}{2}$ to the difference between two consecutive mel spectrogram bands. A total of 40 mel bands are used in this model, achieving 160 stacked input features for each time step. The input features are calculated using a sliding window with a hop length of 441 samples (10 ms). Each hop performed by the sliding window corresponds to one time step in the model.

The model consists of a bidirectional long short-term memory network (LSTM) applied to the described input features. Bidirectional recurrent networks incorporate future context into a network by adding an extra layer for each hidden layer, which will process the input sequence backward, while the other layer processes it forwards. It produces a non-causal model that outputs values that depend on both the past and future time steps. The network contains 6 hidden layers, 3 of them for processing the input forwardly and 3 backwardly. Each hidden layer contains 20 LSTM units, and the output layer has two units, which are normalized to sum up to 1.0, using the softmax function. The outputs represent the probabilities of onset and no onset. The authors reported that they used this two-output approach because the results using a single output were not as successful. To obtain the final onset times, a peak picking method is used to detect the local maxima just for the output corresponding to the onset class.

2.2.3. SuperFlux

One of the most popular DSP-based methods for note onset detection is based on calculating the spectral flux, based on the idea that note onsets are accompanied by rapid spectral changes in the audio signal. The *SuperFlux* method, proposed by Böck and Widmer [9], consists in adding some improvements to the spectral flux method. Instead of calculating the bin-wise difference between consecutive spectral frames, it adds a trajectory-tracking stage to the method to avoid high values in the ODF. It also combines phase information, using a technique called local group delay, to reduce the impact of amplitude variations that occur in steady tones in the output of the method. These changes seek to reduce false positives caused by *vibrato* and *tremolo*.

2.2.4. Clarinet-specific model

To obtain a specialized note onset detector for clarinet recordings, we implemented the CNN architecture described in section 2.2.1 and trained this network on a clarinet dataset. In the original paper, by Schlüter and Böck [7], their model had been trained on a dataset containing both monophonic and polyphonic recordings played on various instruments and covering multiple musical genres. We anticipated that that model would not perform as well as it could on clarinet recordings due to the specific characteristics of the sound produced by the instrument, which are underrepresented in the training data.

We adjusted a few hyperparameters in our model. The learning rate was set to 0.1, and the weights of the frames that precede and succeed the annotated onset were set to 0.4 (fuzziness). These hyperparameters were adjusted by testing different values and evaluating the result on a validation set.

2.3. Evaluation Criteria

To evaluate a note onset detection model, we need to define criteria to decide whether each onset prediction is correct or not. Thus, the predictions must be compared and matched to the onset annotations (ground truth). For this purpose, we specified an argument ω, which is a tolerance window (defined in milliseconds) centered on each annotated value. To be considered valid, a prediction must fall within the defined tolerance window around the onset annotation. When two onset predictions fall within the tolerance window around a single annotation, only the closest one is accounted as a true positive, while the other is considered a false positive. With those values, we can calculate the metrics precision, recall, and F-score, to evaluate the results of an onset detection method. All the results presented in this paper show the evaluation metrics calculated using two different values for the tolerance window ω: 20 ms and 50 ms. Since these tolerance windows are centered on the target onset point, an onset detection that is within their limits will correspond to a maximum distance of 10 ms and 25 ms, respectively.

3. Results

The following names are used to refer to the models we evaluated in our experiments:

- CNN-Böck
- RNN-Böck
- SuperFlux
- CNN-Clari

The models *CNN-Böck* and *RNN-Böck* were pretrained on a generic dataset, known as Böck dataset, containing 102 minutes of audio and 25,927 onsets composed by multiple instruments and musical genres, with both polyphonic and monophonic recordings. The model *Superflux* is based on DSP techniques. *CNN-Clari* was trained by ourselves using the dataset *clari-onsets-50*.

Table 2 shows the results obtained using each of the models on the dataset *clari-onsets-50*. To train and evaluate the model *CNN-Clari* using the same dataset, we employed 10-fold cross-validation, using 8 folds for training, 1 for validation, and 1 for testing. The model *CNN-Clari* generated the best results among the four models,

	$\omega = 50$ ms			$\omega = 20$ ms		
method	f-score	recall	precision	f-score	recall	precision
CNN-Böck	0.861	0.851	0.872	0.605	0.597	0.614
RNN-Böck	0.781	0.779	0.784	0.459	0.456	0.461
SuperFlux	0.718	0.669	0.775	0.379	0.353	0.410
CNN-Clari	0.954	0.946	0.962	0.720	0.712	0.728

Table 2: Results obtained for the dataset *clari-onsets-50*. The model CNN-Clari was trained and evaluated using 10-fold cross-validation, using 8 folds for training, 1 for validation, and 1 for testing.

	$\omega = 50$ ms			$\omega = 20$ ms		
method	f-score	recall	precision	f-score	recall	precision
CNN-Böck	0.948	0.944	0.952	0.709	0.706	0.712
RNN-Böck	0.887	0.905	0.870	0.685	0.698	0.672
SuperFlux	0.862	0.889	0.836	0.469	0.484	0.455
CNN-Clari	0.988	0.984	0.992	0.908	0.905	0.912

Table 3: Results obtained for the dataset *clari-onsets-3*. This time, the model *CNN-Clari* was trained on the entire dataset *clari-onsets-50*, and the resulting model was employed on the recordings from *clari-onsets-3*.

for tolerance windows of both 50 ms and 20 ms, with respect to all the three metrics: F-score, recall, and precision. The F-score obtained for the 50 ms tolerance window was 0.954, more than nine percentual points higher than the second best F-score. Among the models from the library madmom, *CNN-Böck* was the best, reaching an F-score of 0.861 against 0.781 from *RNN-Böck* and 0.718 from *SuperFlux*. This result corroborates with the results obtained in [7] when comparing these three models, in which the CNN had performed better that the other two models, on a generic dataset. Nonetheless, the results

The results obtained for the dataset *clari-onsets-3* are shown in table 3. For this particular experiment, we trained the model *CNN-Clari* using the entire dataset *clari-onsets-50*.

Again, the model *CNN-Clari* produced the best results, with an F-score of 0.988 for the 50 ms tolerance window, followed by *CNN-Böck* (0.948), *RNN-Böck* (0.887) and *SuperFlux* (0.862). The results obtained for this smaller dataset were considerably better than for *clari-onsets-50*.

4. Discussion

Training a CNN model on a clarinet dataset, specifically, we were able to achieve results that are significantly better, for this particular domain, than those produced by models trained on generic data or by a model that uses strictly DSP techniques. If we order the results from the models based on the resulting F-score, we obtain the sequence *CNN-Clari*, *CNN-Böck*, *RNN-Böck* and *SuperFlux*. This sequence is consistent among the results from both experiments and both tolerance windows.

Regarding the dataset *clari-onsets-3*, for which we obtained results that are considerably better when compared to *clari-onsets-50*, it is worth emphasizing that it is a small dataset, having a total duration that corresponds to only 3.4% percent of the latter, and there is certainly a bias related to the fact that its recordings were all made in a laboratory, under controlled conditions, in a room with very low reverberation, by only two clarinetists. The low reverberation is probably a factor that makes it a lot easier for any method to detect the onsets since it minimizes the superposition of energy of consecutive notes.

By listening to the note onsets detected by the clarinet-specific model, we observed that it tended to miss some onset for notes with extremely long attack times (very soft notes), producing false negatives. During the dataset annotation we had already noticed that it was quite hard to determine the exact position of the onset for such notes. Thus, we suspect that those false negatives might have some relation to the annotated data's imprecision for soft notes. Also, on a few situations where there were two consecutive notes with the same pitch, the model tended to miss the second note's onset.

In the clarinet recordings we used, there is a significant amount of breathing sounds, background noises, crackling sounds from the chair, and sometimes even whisper sounds. It is interesting to notice that the network was able to learn to ignore those sounds completely, which is an excellent result, since they would not correspond to note onsets in the vast majority of the clarinet repertoire (although they might, in contemporary compositions).

It is worth mentioning that we did some preliminary experiments training the clarinet onset detection model using additional semi-synthetic data, but so far, the results have not been promising. That data was created using a MIDI keyboard controller to record monophonic melodies using the sound from a commercial clarinet sampler software. The *note on* events of the MIDI protocol were used to generate the onset annotations, and those were used in our training experiments. We attempted to use this semi-synthetic data in the CNN model using two different strategies: (1) employing them on a pre-training phase and (2) mixing them in the training data. None of these approaches

improved the results of the model. As a matter of fact, they produced slightly worse results than the models trained without them. Yet, these results are preliminary and still require further investigation. We also did some experiments applying data augmentation techniques to the train data (such as pitch-shifting, time-stretching, gain adjustment, and addition of noise). Again, in our preliminary experiments using these techniques, the results of the model got slightly worse. One thing that might be worth further investigation are the false negatives produced by the model *CNN-Clari*, specially on notes with soft attack. As shown in tables 2 and 3, the recall for the model *CNN-Clari* is slightly lower than its precision, so if we figure out a way to reduce those false negatives, we may get a significant improvement in the results produced by the model.

Acknowledgements

The development of this work has been supported by CAPES/Brazil (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) and CNPq/Brazil (Conselho Nacional de Desenvolvimento Científico e Tecnológico).

References

[1] Chris Duxbury, Mark Sandler, and Mike Davies. A hybrid approach to musical note onset detection. *Computer*, pages 33–38, 2002.

[2] Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B. Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 13(5):1035–1046, 2005.

[3] Nick Collins. Using a pitch detector for onset detection. *Proceedings of the International Symposium on Music Information Retrieval*, pages 100–106, 2005.

[4] Ruohua Zhou and J.D. Reiss. Music onset detection combining energy-based and pitch-based approaches. *Proc. MIREX Audio Onset Detection Contest*, 2007.

[5] Harvey Thornburg, Randal J. Leistikow, and Jonathan Berger. Melody extraction and musical onset detection via probabilistic models of framewise STFT peak data. *IEEE Transactions on Audio, Speech and Language Processing*, 15(4):1257–1272, 2007.

[6] Sebastian Böck, Filip Korzeniowski, Jan Schlüter, Florian Krebs, and Gerhard Widmer. madmom: a new Python Audio and Music Signal Processing Library. In *Proceedings of the 24th ACM International Conference on Multimedia*, pages 1174–1178, Amsterdam, The Netherlands, 2016.

[7] Jan Schlüter and Sebastian Böck. Improved musical onset detection with Convolutional Neural Networks. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6979–6983, Prague, Czech Republic, may 2014. IEEE.

[8] Florian Eyben, Sebastian Böck, Björn Schuller, and Alex Graves. Universal Onset Detection with Bidirectional Long-Short Term Memory Neural Networks. *Proceedings 11th International Society for Music Information Retrieval Conference, ISMIR 2010*, (June 2017):589–594, 2010.

[9] Sebastian Böck and Gerhard Widmer. Maximum Filter Vibrato Suppression for Onset Detection. In *Proc. of the 16th Int. Conference on Digital Audio Effects (DAFx-13), Maynooth, Ireland, September 2-5, 2013*, pages 1–7, 2013.