Onsets and Velocities: Affordable Real-Time Piano Transcription Using Convolutional Neural Networks

Andres Fernandez
University of Tübingen and IMPRS-IS
Tübingen, Germany
a.fernandez@uni-tuebingen.de

***Abstract*—Polyphonic Piano Transcription has recently experienced substantial progress driven by the application of sophisticated Deep Learning setups and the introduction of new subtasks such as note onset, offset, velocity and pedal detection. In this work, we focus on *onset and velocity* detection, presenting a convolutional neural network with substantially reduced size (∼3.1M parameters) and a simple inference scheme that achieves state-of-the-art performance on the MAESTRO dataset for onset detection (F_1=96.78%) and sets a good novel baseline for onset+velocity (F_1=94.50%), while maintaining real-time capabilities on modest commodity hardware. Furthermore, our proposed ONSETS&VELOCITIES (O&V) model shows that a time resolution of 24ms is competitive, countering recent trends. We provide open-source software to reproduce our results and a real-time demo with a pretrained model [1].**

***Index Terms*—deep learning, polyphonic piano transcription**

I. INTRODUCTION

A. Polyphonic Piano Transcription

The task of *Polyphonic Piano Transcription* (PPT) is useful for downstream tasks like musical analysis and resynthesis. Consider an audio *waveform* $x(t) \in \mathbb{R}^T$ with time T that corresponds to a piano performance of a *score* $\mathcal{S}$; then the task of PPT is to recover $\mathcal{S}$ from x. Here, $\mathcal{S}$ is a collection of N *note events* $\{\mathcal{N}_n := (k_n, v_n, \downarrow_n, \uparrow_n)\}_{n=1}^{N}$, where $k \in \{1, \dots, K\}$ specifies the *key* (typically $K = 88$). The value $v \in [0, 1]$ indicates the intensity of the event (also called key *velocity*). The key *onset* (pressing) and *offset* (releasing) timestamps are specified by $\downarrow$ and $\uparrow$, respectively, where $0 \leq \downarrow_n < \uparrow_n \leq T \quad \forall n$.

There has been extensive effort in automating PPT, typically articulated through challenges like the popular *Music Information Retrieval Evaluation eXchange* (MIREX) [1] and featuring different techniques like handcrafted features, spectrogram factorization, probabilistic models [2], [3] and, more recently, *Deep Learning* (DL) [4]. PPT is typically evaluated by comparing the recovered score $\hat{\mathcal{S}}$ with the ground truth $\mathcal{S}$ on a test set, in an event-wise manner. Prominent efforts in curating datasets like MAPS [5], [6], SMD [7] and MusicNet [8] were affected by imprecise annotations, insufficient training data, unrealistic interpretations and/or constrained recording conditions, which made evaluation more difficult and impeded the establishment of a unified benchmark for PPT [9]. The introduction of the MAESTRO dataset [9] addressed many of these issues, by providing ∼200 hours of precisely annotated, high-quality audio data, encompassing a large variety of virtuosistic compositions, pianists and recording conditions, and incorporating evaluation splits. As a result, it quickly became a popular benchmark. Still, all pianos in MAESTRO are fairly similar: to capture more general settings and satisfy the ever-growing demand for training data, [10] curated the GiantMIDI dataset, by sourcing over 1000 hours of piano music from YouTube and annotating them using DL [11].

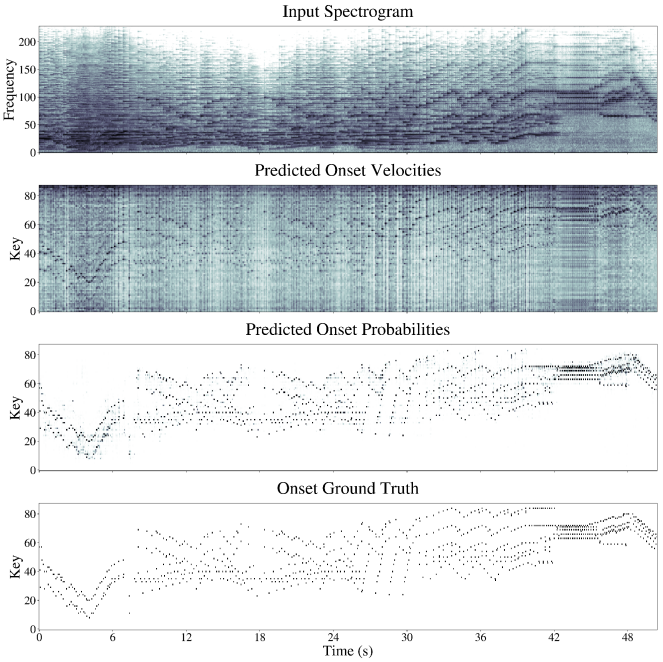


Fig. 1: Log-mel spctrogram (X) of a virtuosistic excerpt from the MAESTRO test set (Franz Liszt, *Concert Étude "Waldesrauschen"*), followed by the corresponding velocity ($\hat{\mathcal{R}}_V$) and last-stage onset ($\hat{\mathcal{R}}_{\downarrow}^{(3)}$) predictions, as well as the ground truth *piano roll* $\mathbb{1}_{3\downarrow}$ (see Section II for details).

B. The State of the Art in PPT

One influential effort applying DL to PPT was [12]. Their work cemented the following main trends:

a) Spectrograms: Despite promising efforts to use the $x(t)$ waveforms as DL inputs [13], variants of the spectrogram

Work done as independent researcher at the IAMÚSICA project supported by the *Institut d'Estudis Balearics*, Balearic Islands.

[1] Setup https://github.com/andres-fr/iamusica_training
Demo: https://github.com/andres-fr/iamusica_demo
IAMÚSICA: https://joantrave.net/en/iamusica/

[14, ch.19] $|\int_{-\infty}^{\infty} x(\tau)w(\tau-t)e^{-if\tau}d\tau| \in \mathbb{R}^{F\times T}$ (see Figure 1) remain competitive for PPT and discriminative audio tasks in general [11], [15].

b) Piano roll supervision: Consider an alternative representation of $\mathcal{S}$, called *piano roll* $\mathcal{R} \in [0,1]^{K\times T}$ (see Figure 1), where entries $\mathcal{R}(k,t)$ encode the activity of channel k at time t (zero if inactive). This type of supervision consists in training the model to output a piano roll $\hat{\mathcal{R}}$ that predicts $\mathcal{R}$, by minimizing the binary cross-entropy loss: $l_{BCE}(\mathcal{R},\hat{\mathcal{R}}) = \langle \mathcal{R}, -log(\hat{\mathcal{R}})\rangle + \langle (1-\mathcal{R}), -log(1-\hat{\mathcal{R}})\rangle$. If the ground truth is boolean, we instead minimize $l_{BCE}(\mathbb{1},\hat{\mathcal{R}})$, where $\mathbb{1} \in \{0,1\}^{K\times T}$ is a binarized piano roll. This approach requires to *decode* the predicted piano roll $\hat{\mathcal{R}}$ to obtain the event-based representation $\hat{\mathcal{S}} = dec_H(\hat{\mathcal{R}})$, typically by using a heuristic H, e.g. grouping consecutive active frames into singe notes.

c) Computer Vision: PPT can be tackled effectively by treating spectrograms and piano rolls as images, and models like Convolutional Neural Networks (CNNs) [16] work well with minor adaptions.

A major turning point was ONSETS&FRAMES (O&F) [17], which uses a sub-network to first predict a piano roll $\hat{\mathcal{R}}_{\downarrow}$ encoding the probability of an *onset* (i.e. $\mathbb{1}_{\downarrow}$ that is active at the moment a key is pressed), and then uses another subnetwork to predict $\hat{\mathcal{R}}_{\mathcal{N}}$ conditioned on $\hat{\mathcal{R}}_{\downarrow}$, encoding the probability of a *frame* (i.e. $\mathbb{1}_{\mathcal{N}}$, active for the whole duration of each note). O&F is then trained jointly via a multi-task loss $l_{BCE}(\mathbb{1}_{\downarrow},\hat{\mathcal{R}}_{\downarrow}) + l_{BCE}(\mathbb{1}_{\mathcal{N}},\hat{\mathcal{R}}_{\mathcal{N}})$. O&F achieved a steep improvement in all PPT benchmarks, and also introduced a novel subtask, *note velocity*, modelled with a third sub-network that predicts a velocity piano roll $\hat{\mathcal{R}}_V$ trained via masked ℓ_2-norm loss $l_V = \langle \mathbb{1}_{\downarrow}, (\mathcal{R}_{\mathcal{N}} - \hat{\mathcal{R}}_V)^2\rangle$. Due to its unprecedented effectiveness and versatility, O&F became a popular baseline [18], [19], but this was at the expense of increased complexity, including more elaborate decoder heuristics, a larger model, and the incorporation of bi-RNN layers [20], [21], which preclude real-time applications.

More recently, [11] pointed out issues with temporal precision on piano rolls, incorporating a trainable REGRESSION model to enhance precision. They further expanded the model including *sustain pedal detection* capabilities. In [22], an *off-the-shelf* TRANSFORMER [23] setup was used to predict $\hat{\mathcal{N}}$ directly from spectrograms. Apart from their good performance, both systems have in common their substantial size, increased time resolution and replacing decoder heuristics with an end-to-end differentiable solution, suggesting that decoding is a performance bottleneck.

C. Proposed Contribution for PPT

These state-of-the-art improvements in performance came entangled with increased complexity in the form of larger models, additional components and new sub-tasks [19]. Understanding and disentangling this complexity is an active field of research: Alternative PPT sub-task factorizations that do not rely on O&F were proposed, like nonlinear denoising vs. linear demixing [24], sound source vs. note arrangement [25] and ADSR envelopes [26]. General approaches like using invertible neural networks [27], reconstruction tasks [28] and additive attention [19] were also explored.

In this work, we pursue the orthogonal goal of achieving real-time capabilities. For that, we observe that the masked loss l_V imposes time-locality around the onsets, and follow up on several ideas: the importance of the onsets [17] as well as decoder heuristics [19], and the idea that note velocity is naturally associated with the onset [11]. We propose that *a convolutional end-to-end method for onsets and velocities leads to efficiency gains and affordable real-time capabilities without compromising performance*, and that *efficient decoding heuristics replace the need for high temporal resolution and complex inference schemes*.

We present ONSETS&VELOCITIES (O&V), featuring:

1) State-of-the-art performance for *onset* detection and a good baseline for *onsets+velocities* on MAESTRO.
2) A substantially smaller CNN and a simple decoder, enabling affordable real-time inference on CPU.
3) A multi-task training scheme without any data augmentations or extensions, based on piano rolls at 24ms resolution.

In Section II we present our O&V method. Section III presents experiments substantiating our claims. We also provide a PyTorch [29] open-source implementation with a real-time demo. Section IV concludes and proposes future work.

II. METHODOLOGY

A. Model

Given a waveform $x(t) \in \mathbb{R}^T$ at 16kHz, we compute its STFT [14] with a Hann window of size 2048, and a hop size δ=384 (i.e. a time resolution of Δ_t=24ms). We then map it to 229 mel-frequency bins [30] in the 50Hz-8000Hz range, and take the logarithm, yielding our input representation: a *log-mel spectrogram* $X(f,t') \in \mathbb{R}^{229\times T'}$, where $T' = \frac{T}{\delta}$ is the resulting "compact" time domain (see Figure 1). We also compute the first time-derivative $\dot{X}(f,t') := X(f,t') - X(f,t'-1)$ and concatenate it to X, forming the CNN input. Using the same Δ_t, we time-quantize the MIDI annotations into a piano roll $\mathcal{R}_V \in [0,1]^{88\times T'}$, where $\mathcal{R}_V(k_n, t'_n)$ contains the velocity if key k_n was pressed at time $\Delta_t t'_n \pm \frac{\Delta t}{2}$, and zero otherwise. We further binarize $\mathcal{R}_V$, yielding $\mathbb{1}_{\downarrow}$.

The complete CNN is presented in Figure 2. We highlight the following design principles:

a) No recurrent layers: Motivated by [31], [32], we follow the established CNN design of stem, body and head, making use of residual bottlenecks [33].

b) No pooling: Motivated by [34], all residual bottlenecks maintain activation shape, and conversion from input to output shape is done in a single depthwise convolution layer [35], shown to be efficient and effective [36]. Note that the convolutions in the input domain (spectrogram) have vertical dimensions but the convolutions in the output domain (piano

roll) do not, since we assume that neighbouring frequencies are related but neighbouring piano keys aren't.

c) Multi-stage: Inspired by OPENPOSE [37], O&V features a series of residual stages that sequentially refine and produce the output. This is useful for real-time applications, since stages can be easily removed without need for retraining.

d) Temporal context: At its core, O&V features the *Context-Aware Module* (CAM) [38], which is a residual bottleneck that combines time-dilated convolutions and channel attention [39]. Inspired by TCNs [40] and INCEPTION [41], we aim to capture the temporal vicinity of an onset efficiently.

e) Model regularizers: At the input and before each output, O&V features *Sub-Spectral Batch Normalization* (SBN), i.e. one individual BN per vertical dimension [42], [43]. We add dropout [44], [45] after the parameter-heavy layers. We use leaky ReLUs [46], [47] as nonlinearities.

f) Time locality: The time-derivative $\dot{X}$ is a handcrafted input feature that directly represents intensity variations.

During inference, O&V produces one piano roll per onset stage $(\hat{\mathcal{R}}_\downarrow^{(1)}, \hat{\mathcal{R}}_\downarrow^{(2)}, \dots)$ and one velocity piano roll $\hat{\mathcal{R}}_{\downarrow V}$ (see Figure 2(d)). Then, our proposed decoder $dec_{\sigma,\rho,\mu}(\hat{\mathcal{R}}_\downarrow, \hat{\mathcal{R}}_V)$ follows a simple heuristic: temporal Gaussian smoothing (*smooth*) with variance σ^2 followed by non-maximum suppression (*nms*), thresholding ρ and shifting μ:

$$\begin{aligned} \hat{\downarrow} &:= \big\{(k,t') :\ nms\big(smooth_\sigma(\hat{\mathcal{R}}_\downarrow)\big)_{(k,t')} \geq \rho\big\} \\ \hat{\mathcal{S}} &:= \{(k_n,\ \hat{\mathcal{R}}_V(k_n,t'_n),\ \Delta_t t'_n + \mu) :\ (k_n,t'_n) \in \hat{\downarrow}_\rho\} \end{aligned} \tag{1}$$

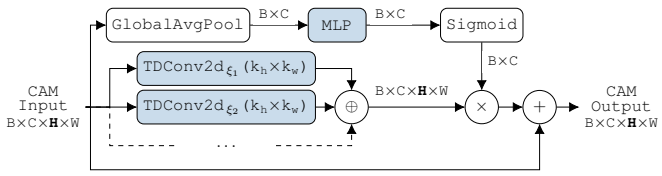
The note events are read at the resulting locations, and shifted by a global constant μ. In this work, we use the last (third) onset stage $(\hat{\mathcal{R}}_\downarrow^{(3)})$ and the values $\sigma = 1, \mu = -0.01s, \rho = 0.74$, obtained via cross-validation of the trained CNN on a subset of the MAESTRO validation split (note that this is different from the test split used for evaluation). While the optimal ρ fluctuates during training, we found $\sigma = 1,\ \mu = -0.01s$ to be stable.

B. Training

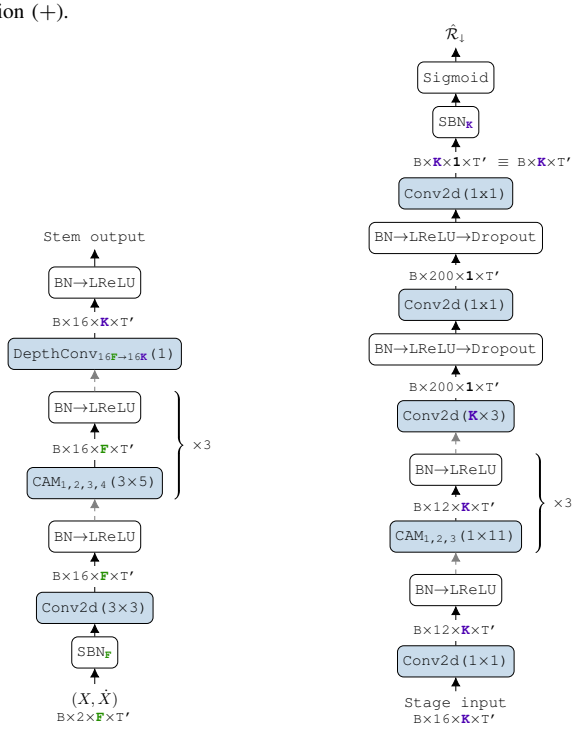
We train our CNN to predict onset probability and velocity jointly via minimization of the following multi-task loss:

$$\begin{aligned} l_{\downarrow V}(\mathbb{1}_\downarrow, (\hat{\mathcal{R}}_\downarrow^{(1)}, \hat{\mathcal{R}}_\downarrow^{(2)}, \dots), \mathcal{R}_V, \hat{\mathcal{R}}_V) := \\ \lambda_1 \cdot \sum_i l_{BCE}(\mathbb{1}_{3\downarrow}, \hat{\mathcal{R}}_\downarrow^{(i)}) + \lambda_2 \cdot l_{V'}(\mathcal{R}_V, \hat{\mathcal{R}}_V) \\ \text{where } l_{V'}(\mathcal{R}_V, \hat{\mathcal{R}}_V) := \\ \big\langle \mathbb{1}_\downarrow,\ \big(\mathcal{R}_V \cdot -log(\hat{\mathcal{R}}_V)\big)\big((1-\mathcal{R}_V)\cdot - log(1-\hat{\mathcal{R}}_V)\big)\big\rangle \end{aligned}$$

The masked loss $l_{V'}$ is a cross-entropy variant of l_V introduced in [17] and [11] that encourages to predict the right velocity only in the vicinity of onsets. The $\mathbb{1}_{3\downarrow}$ mask consists in modifying $\mathbb{1}_\downarrow$ by adding 2 further active frames after each onset (i.e. onset labels span 3 frames instead of 1). This simple extension, combined with our decoder, allows for effective training and inference, bypassing the need for elaborate decoding schemes as the ones discussed in [11].

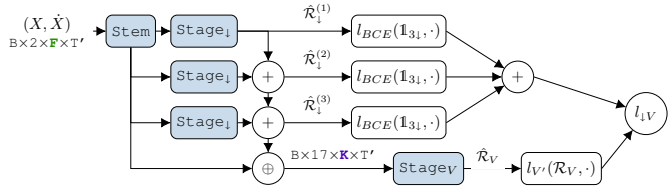


(a) Diagram of a **CAM**$_{\xi_1,\xi_2,\dots}(k_h \times k_w)$, based on [38]. The middle branch concatenates ($\oplus$) multiple time-dilated convolutions (TDConv), each with time dilation ξ. The output channels of each TDConv is **H** divided by the number of TDConv layers being concatenated, and padding is adjusted so shape is preserved. The top branch acts as a channel-wise attention mechanism ($\times$), featuring a Multi-Layer Perceptron (MLP) acting as a bottleneck (we use 2 layers with ReLU activation and 8 hidden dimensions). The bottom branch is a residual connection ($+$).



(b) Diagram of the **Stem**, which is a CAM-powered residual CNN. The SBN has **F** frequency bands (one per vertical dimension). The ×3 braces indicate 3 sequential blocks. Note the single-step **F**→**K** transition via depthwise convolution DepthConv(1) with temporal kernel width of 1.

(c) Diagram of an onset **Stage**$_\downarrow$. It is a modification of the stem, followed by convolutions that act like an MLP moving across time dimension T′. We add dropout to the MLP. The velocity **Stage**$_V$ is like Stage$_\downarrow$, except it has only {×1} CAM blocks and 17 input channels instead of 16.



(d) Diagram of our proposed 3-stage ONSETS&VELOCITIES full architecture and training loss. Each Stage$_\downarrow$ produces an onset piano roll $\hat{\mathcal{R}}_\downarrow$, and successive stages refine the output via the residual connection ($+$ preserves shape). The velocity Stage$_V$ uses knowledge about onset locations from the final onset stage, and everything is trained jointly. See Figure 1) for qualitative examples of X, $\hat{\mathcal{R}}_\downarrow^{(3)}$ and $\hat{\mathcal{R}}_\mathcal{N}$. Section II-B presents the loss $l_{\downarrow V}$.

Fig. 2: Our proposed CNN. Rank-4 tensor dimensions are Batch×Channel×Height×Width.

TABLE I: Comparison of top-performing models in terms of specifications (number of parameters, architecture and functionality) and performance (precision, recall, F_1-score and MAESTRO version).

Model	Onset+Velocity # Params	Architecture	Offset/Pedal?	Onset (%)			Onset+Velocity (%)			MAESTRO version
				P	R	F_1	P	R	F_1	
O&F [17]	10M	bi-RNN	✓ / ✗	98.27	92.61	95.32	-	-	-	v1
Regression [11]	12M	bi-RNN	✓ / ✓	98.17	95.35	96.72	-	-	-	v2
Transformer [22]	–	Transformer	✓ / ✗	-	-	96.13	-	-	-	v3
O&V (ours)	**3.13M**	CNN	✗ / ✗	98.58	95.07	**96.78**	96.25	92.86	94.50	v3

All model weights are initialized with the Gaussian-He distribution [48], and biases with 0, except the CAM channel atention biases (right before the sigmoid), initialized with 1 to promote signal flow. We use the Adam optimizer with a decoupled weight decay [49], [50] of 3×10^{-4}, trained with random batches of 5-second segments (batch size 40, ∼14k batches per epoch) for ∼70K batches. For the learning rate, we start with a ramp-up from 0 to 0.008 across 500 batches, followed by cosine annealing with warm restarts [51], using cycles of 1000 batches and decaying by 97.5% after each cycle. BN/SBN momentum is 95%, dropout 15% and leaky ReLUs have a slope of 0.1. In l'_V, we use $\lambda_1 = 1, \lambda_2 = 10$. To compensate that $\mathbb{1}_{3\downarrow}$ is sparse, we give positive entries a weight 8 times bigger than negative entries inside of $l_{BCE}(\mathbb{1}_{3\downarrow}, \cdot)$. Training speed was 1800 batches per hour on a 2080Ti NVIDIA GPU.

C. Evaluation

Following the same evaluation procedure as O&F [17], Regression [11] and Transformer [22], and applying standard metrics from [52] implemented in the `mir_eval` library [53], we report precision (P), recall (R) and F_1-score for the predicted *onsets*, considered correct if they are within 50ms of the ground truth. The *onset+velocity* evaluation, following O&F [17, 3.1], has an added constraint: the predicted velocity must also be within 0.1 of the ground truth normalized between 0 and 1.

Note that the MAESTRO dataset is being actively extended and curated, presenting 3 versions so far. We report the respective versions in Table I, noting that versions 2 and 3 are almost identical, although comparisons across versions should be taken approximately.

III. Experiments and Discussion

We trained O&V on the MAESTRO v3 training split without any extensions or augmentations, achieving state-of-the-art performance in onset detection (see Table I). In the following we discuss some implications:

a) Temporal resolution: Our results seem to counter the need for increased temporal resolution expressed in [11] (which use 8ms), showing that 24ms piano rolls coupled with our decoder presented in Equation (1) are competitive.

b) Reduced memory footprint: Table I provides model parameters that are responsible exclusively for onset and frame detection (Transformer has ∼54M parameters in total, but it transcribes everything jointly so it cannot be fairly compared). O&V outperforms the best alternative, Regression, with ∼4 times less parameters.

c) Affordable real-time inference: Bi-recurrent layers like the ones used in O&F and Regression are unsuited for real-time processing. Transformer took ∼380s to transcribe a 120s file on an Intel Xeon CPU (1 core), and ∼20s on a Tesla-T4 GPU (including offsets) when run on the official Colab implementation[2]. O&V took less than 2s to process the same file on an 8-core Intel i7-11800H CPU. Even accounting for the number of cores, O&V is approximately one order of magnitude faster than Transformer.

d) Conceptual simplicity: In essence, O&V revives the simplicity from [12] by applying a feedforward CNN to a discriminative task via piano rolls, followed by a simple decoding heuristic. Its architecture allows to remove onset stages without retraining, providing a flexible trade-off between runtime and performance with little added complexity.

e) Latency: The receptive field for our O&V proposed components is: `Stem`: 60 frames (1.44s), $\texttt{Stage}_{\downarrow}$: 99 frames (∼2.38s), and $\texttt{Stage}_{\texttt{V}}$: 35 frames (0.84s). This would theoretically impose a latency of over 9s, which is far from a responsive system. We informally note that the latency can be truncated without drastically affecting results (we used a latency of 4s in a live workshop), and encourage practical applications. We also note that our focus was on finding a CNN with affordable inference and competitive performance, and we did not optimize for low receptive field, which may be obtainable with minor variations to the architecture (e.g. reducing the number of consecutive stages or CAMs).

IV. Conclusion and Future Work

We presented Onsets&Velocities, a CNN with a training scheme based on piano rolls with 24ms of resolution and a straightforward decoding heuristic. O&V achieves state-of-the-art performance in PPT note *onset* detection, and establishes a good baseline on *onset+velocity* detection, while significantly reducing model size and inference complexity, and offering affordable and flexible real-time capabilities. Future work could include reducing the receptive field, extensions to *offset* and *pedal* detection, evaluation using less pre-trained stages and analysis of design choices via ablation studies.

Acknowledgments

A.F. wants to thank Jesús Monge Álvarez and Christian J. Steinmetz for their valuable feedback, the *Institut d'Estudis Baleàrics* for supporting this work with research grant 389062 INV-23/2021, and the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for further support.

[2]https://github.com/magenta/mt3

REFERENCES

[1] J. Downie, D. Byrd, and T. Crawford, "Ten years of ISMIR: Reflections on challenges and opportunities." *ISMIR Proc.*, pp. 13–18, 01 2009.

[2] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription: Challenges and future directions," *Journal of Intelligent Information Systems*, vol. 41, 12 2013.

[3] J. Driedger, "Processing music signals using audio decomposition techniques," doctoral thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2016.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–44, 05 2015.

[5] V. Emiya, N. Bertin, B. David, and R. Badeau, "MAPS - a piano database for multipitch estimation and automatic transcription of music (research report)," 2010.

[6] A. Ycart and E. Benetos, "A-MAPS: Augmented MAPS dataset with rhythm and key annotations," *19th ISMIR Conference*, 09 2018.

[7] M. Müller, V. Konz, W. Bogler, and V. Arifi-Müller, "Saarland music data (SMD)," *Late-Breaking and Demo Session of the 12th ISMIR*, 2011.

[8] J. Thickstun, Z. Harchaoui, and S. M. Kakade, "Learning features of music from scratch," in *ICLR*, 2017.

[9] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the MAESTRO dataset," in *ICLR*, 2019. [Online]. Available: https://openreview.net/forum?id=r1lYRjC9F7

[10] Q. Kong, B. Li, J. Chen, and Y. Wang, "GiantMIDI-Piano: A large-scale MIDI dataset for classical piano music." *ISMIR Trans.*, pp. 87–98, 2022.

[11] Q. Kong, B. Li, X. Song, Y. Wan, and Y. Wang, "High-resolution piano transcription with pedals by regressing onset and offset times," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 29, p. 3707–3717, oct 2021.

[12] R. Kelz, M. Dorfer, F. Korzeniowski, S. Böck, A. Arzt, and G. Widmer, "On the potential of simple framewise approaches to piano transcription," in *ISMIR Proceedings*, august 2016, pp. 475–481.

[13] N. Zeghidour, O. Teboul, F. de Chaumont Quitry, and M. Tagliasacchi, "LEAF: A learnable frontend for audio classification," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=jM76BCb6F9m

[14] R. Bracewell, *The Fourier Transform and its Applications*, 2nd ed. Tokyo: McGraw-Hill Kogakusha, Ltd., 1978.

[15] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[16] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, *Object Recognition with Gradient-Based Learning*. Springer Berlin Heidelberg, 1999, pp. 319–345. [Online]. Available: https://doi.org/10.1007/3-540-46805-6_19

[17] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. H. Engel, S. Oore, and D. Eck, "Onsets and frames: Dual-objective piano transcription." in *ISMIR*, 2018, pp. 50–57.

[18] J. Kim and J. Bello, "Adversarial learning for improved onsets and frames music transcription," in *ISMIR Proceedings*, 2019, pp. 670–677.

[19] K. Cheuk, Y. Luo, E. Benetos, and D. Herremans, "Revisiting the onsets and frames model with additive attention," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2021.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[21] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. on Sig. Proc.*, vol. 45, no. 11, pp. 2673–2681, 1997.

[22] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. H. Engel, "Sequence-to-sequence piano transcription with transformers," in *ISMIR Proceedings*, november 2021, pp. 246–253.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[24] R. Kelz and G. Widmer, "Nonlinear denoising, linear demixing," in *I (Still) Can't Believe It's Not Better! NeurIPS 2021 Workshop*, 2021. [Online]. Available: https://openreview.net/forum?id=CnsnlBlkCXx

[25] L. S. Marták, R. Kelz, and G. Widmer, "Balancing bias and performance in polyphonic piano transcription systems," *Frontiers in Signal Processing*, vol. 2, 2022.

[26] R. Kelz, S. Böck, and G. Widmer, "Deep polyphonic ADSR piano note transcription," *ICASSP*, pp. 246–250, 2019.

[27] R. Kelz and G. Widmer, "Towards interpretable polyphonic transcription with invertible neural networks," in *ISMIR Proceedings*, 2019.

[28] K. W. Cheuk, Y.-J. Luo, E. Benetos, and D. Herremans, "The effect of spectrogram reconstruction on automatic music transcription: An alternative approach to improve transcription accuracy," 10 2020.

[29] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, pp. 8024–8035.

[30] S. S. Stevens and J. E. Volkmann, "The relation of pitch to frequency: A revised scale," *American Journal of Psychology*, vol. 53, p. 329, 1940.

[31] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollar, "Designing network design spaces," in *CVPR Proceedings*, June 2020.

[32] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," *CVPR PRoceedings*, 2022.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *CVPR Proceedings*, Jun. 2016, pp. 770–778.

[34] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller, "Striving for simplicity: The all convolutional net," in *ICLR*, 2015.

[35] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *CVPR Proceedings*, 2017, pp. 1800–1807.

[36] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, 2017.

[37] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh, "OpenPose: Realtime multi-person 2d pose estimation using part affinity fields," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2019.

[38] J. Zhang, Z. Chen, and D. Tao, "Human keypoint detection by progressive context refinement," 10 2019. [Online]. Available: https://arxiv.org/abs/1910.12223

[39] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *CVPR Proceedings*, 2018, pp. 7132–7141.

[40] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling." *CoRR*, 2018. [Online]. Available: http://arxiv.org/abs/1803.01271

[41] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR Proceedings*, 2015, pp. 1–9.

[42] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML Proceedings*, 2015, p. 448–456.

[43] S. Chang, H. Park, J. Cho, H. Park, S. Yun, and K. Hwang, "Subspectral normalization for neural audio data processing," *ICASSP*, 2021.

[44] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[45] X. Li, S. Chen, X. Hu, and J. Yang, "Understanding the disharmony between dropout and batch normalization by variance shift," in *CVPR Proceedings*, 2019, pp. 2682–2690.

[46] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML Proceedings*, 2010, p. 807–814.

[47] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *ICML Proceedings*, 2013.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *ICCV Proceedings*, 2015, pp. 1026–1034.

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR Proceedings*, 2015.

[50] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *ICLR Proceedings*, 2019.

[51] Ilya Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *ICLR Proceedings*, 2017.

[52] M. Bay, A. F. Ehmann, and J. S. Downie, "Evaluation of multiple-F0 estimation and tracking systems," in *ISMIR Proc.*, 2009, pp. 315–320.

[53] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, "`mir_eval`: A transparent implementation of common MIR metrics," in *ISMIR Proceedings*, 2014, pp. 367–372.