A Divide and Conquer Approach to Automatic Music Transcription Using Neural Networks

André Gil[1], Carlos Grilo[1,2(✉)], Gustavo Reis[1,2], and Patrício Domingues[1,2,3]

[1] School of Technology and Management, Polytechnic Institute of Leiria, Leiria, Portugal
2151630@my.ipleiria.pt, {carlos.grilo, gustavo.reis, patricio.domingues}@ipleiria.pt
[2] CIIC, Polytechnic Institute of Leiria, Leiria, Portugal
[3] Instituto de Telecomunicações, Lisbon, Portugal

Abstract. This paper describes a new approach for the automatic music transcription problem. We take advantage of the *divide and conquer* design paradigm and create several artificial neural networks, each one responsible for transcribing one musical note. This way, we depart from the traditional approach which resorts to a single classifier for transcribing all musical notes. To further improve results, an additional post-processing stage using artificial neural networks with the same design paradigm is also proposed. This last stage comprises three main steps: (1) *fix notes duration*, (2) *fix notes duration regarding onsets* and (3) *fix onsets*. The obtained results show that these steps were essential to improve the final transcription. We also compare our results with existing neural network-based approaches. Our approach is able to surpass current state-of-the-art works in frame-based results and, at the same time, reach similar results in onset only, thus demonstrating its viability.

Keywords: Automatic music transcription · Multi-pitch estimation · Artificial neural networks

1 Introduction

Automatic music transcription (AMT) consists in detecting the notes being played in a musical piece, via a machine. This problem is comprised of several sub-problems, which makes a solution for it hard to find. In this work, we mainly focus on the variant called multi-pitch estimation. Multi-pitch estimation consists in identifying the pitched notes present in a polyphonic musical piece. A common approach to this problem is to split a musical piece into smaller chunks, referred to as *frames*, and then, estimate the pitch(es) present in each frame (see Fig. 1).

We apply artificial neural networks (ANNs) to tackle the multi-pitch estimation problem. ANNs have been applied in several different types of problems as, for example, object recognition, image segmentation, speech recognition, text-to-speech synthesis, and, also, music transcription [1–4].

P. Moura Oliveira et al. (Eds.): EPIA 2019, LNAI 11805, pp. 220–231, 2019.
https://doi.org/10.1007/978-3-030-30244-3_19

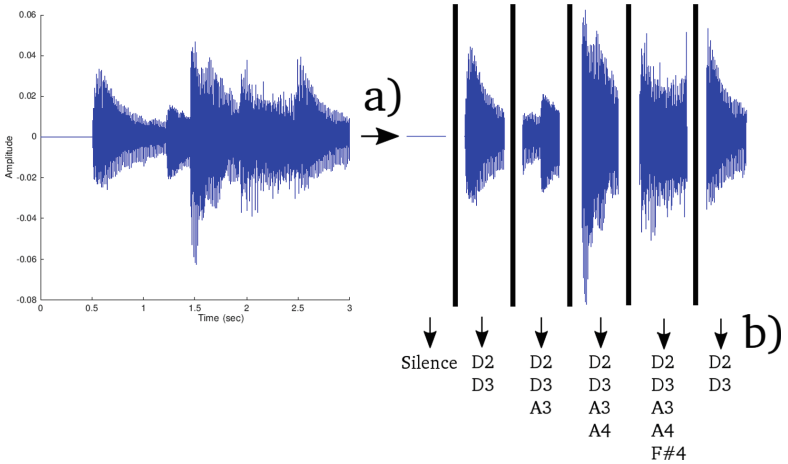

Fig. 1. A common approach to tackle the multi-pitch estimation problem. (a) A musical piece is split into frames. (b) The pitch(es) is estimated for each frame.

The traditional approach to the AMT problem, especially when ANNs are applied, consists in having a single module/network that is responsible for detecting and transcribing all the musical notes in each frame (see Fig. 2a) [2–4]. In this work we use a *divide and conquer* approach which translates into using several networks, referred to as *classifiers*, each one responsible for detecting and transcribing one musical note only (see Fig. 2b). This approach aims at dividing the AMT problem into smaller subproblems, hopefully, easier to solve, possibly boosting the performance of the whole AMT system.

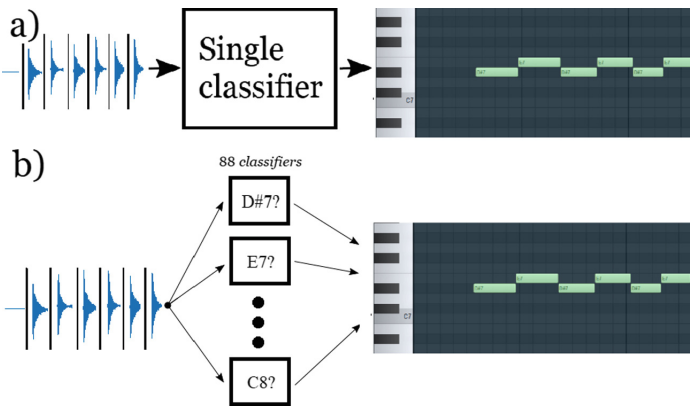

Fig. 2. Representation of (a) the traditional approach for the AMT problem, where a single classifier is responsible for transcribing all musical notes and (b) the *divide and conquer* approach, where several classifiers are used, each one responsible for identifying only one note.

As the pitch estimation process is far from perfect, errors are common. Specifically, two types of errors may arise: (i) musical notes that are not present in a frame are identified as being there and/or, conversely, (ii) notes that are in a frame are not identified. To reduce these types of errors, post-processing methods can be applied. In this work, we propose additional ANNs for that purpose, again following a *divide and conquer* approach.

Some previous works [5, 6] have already applied the *divide and conquer* approach, however, in none of them a comparison with the traditional approach was presented, using the same setup: same dataset and/or same techniques. In this work a comparison is performed between the *divide and conquer* paradigm and the traditional one, using the same dataset, as well as, the same artificial techniques.

The rest of the paper is structured as follows: Sect. 2 describes related work. Section 3 presents our model and Sect. 4 presents and compares the results with other state-of-the-art works. Conclusions and future work are given in Sect. 5.

2 Related Work

Since the first polyphonic music transcription system [7], several approaches have been presented. In 1992, Lea [8] proposed a method that iteratively extracted the predominant peaks. In 2000, Bello and Sandler [9] proposed a simple polyphonic music transcription system using a *blackboard system*. In 2003, Klapuri et al. [10] introduced an algorithm based on harmonicity and spectral smoothness. Also, in 2003 [11], the non-negative matrix factorization technique was introduced for the first time to the AMT problem. In 2004, Moorer [5] introduced for the first time the *divide and conquer* design paradigm to the AMT field, using Artificial Neural Networks. In 2007, Emiya et al. [12] designed a multi-pitch estimation system based on the likelihood maximization principle. In 2008, [13] Yeh proposed a frame-based system to estimate multiple fundamental frequencies of polyphonic music signals. In 2012, Reis et al. [14] introduced for the first time a combination of genetic algorithms with an onset detection algorithm. In 2016, Leite et al. [6] pioneered the coupling of Cartesian Genetic Algorithms with the AMT field, also relying on the *divide and conquer* paradigm. Also in 2016 [2], Convolutional Neural Networks were first introduced to the AMT problem, combined with a *complex* language model to improve their results. In 2016, Kelz et al. [3] proposed a simpler approach to AMT using solely Convolutional Neural Networks. Finally, in 2018, Hawthorne et al. [4] proposed a system that comprises an onset detector and a multi-pitch estimator based on ANNs.

3 Proposed Model

The proposed model consists in a supervised learning system based on several ANNs, each one responsible for transcribing one musical note, resulting in a total of 88 ANNs per dataset, corresponding to the keys in a grand piano. In this work, we have applied classic Multi-Layer Perceptron Neural Networks, instead of more recent techniques as the ones used in Deep Learning, in order to get baseline results. The model is comprised

by three sequential main stages: (1) pre-processing, (2) classification and (3) post-processing. In the following sections, a deeper explanation of each is presented.

3.1 Pre-processing

The pre-processing stage is responsible for splitting the musical pieces into frames and also for converting each frame to the frequency domain, using the Fast Fourier Transform (FFT). Although each frame is comprised of 4096 samples, only the first half is taken into account since the second half of the signal mirrors the first half.

Regarding the ANNs training process, a fundamental key point to consider is the quality of the data. Hence, this stage is responsible for applying two additional sequential transformations used solely in the training set: (i) removal of meaningless data, such as frames with silence and (ii) adapting the ratio between frames with and without the note that should be identified by each classifier, more specifically, 20% of frames with the given note and 80% frames without it (see Fig. 3).

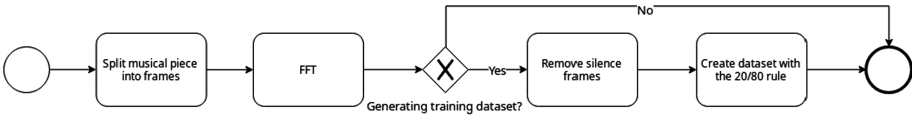

Fig. 3. Transformations applied during the pre-processing stage.

3.2 Classification

The classification stage is where the actual transcription process starts. The resulting data from the pre-processing stage is inserted into this stage so that the note can be detected. As already mentioned, we use the divide and conquer approach, thus, 88 classifiers were created, each one responsible for transcribing one note (see Fig. 2b).

Each classifier comprises five hidden layers with 256, 128, 64, 32 and 8 units, respectively, and an output layer with one unit (*yes* or *no* output). The hidden layers apply the *leaky relu* activation function, while the output layer uses the *sigmoid* function. During the training phase, the optimizer chosen was *Adam* [15] combined with a *learning rate* of 1×10^{-6} and the *cross-entropy* loss function. Also, the following optimization techniques were applied: *data shuffling* [18]; *dropout* [16] with a probability of 0.15; *noisy gradients* [17] with a probability of 0.70 and a standard deviation of 0.05.

3.3 Post-processing

As mentioned earlier, post-processing methods can be applied to correct errors from the classification process. In this work, we use three different types of post-processing methods, labeled as: (i) step 1 - fix notes duration, (ii) step 2 - fix notes duration according to onsets and (iii) step 3 - fix notes onset. Each type applies the *divide and conquer* approach, where an ANN is created to post-process one musical note only, resulting on 88 ANNs per post-processing step. This means that the whole post-

processing stage comprises $88 \times 3 = 264$ ANNs. In the topics that follow, we detail each post-processing type.

Step 1 - Fix notes duration

Music is a time-series phenomenon. By this, we mean that a given event is closely related to a previous and/or a following one. The first post-processing step aims at incorporating that sense of time in the transcription process. Thus, each ANN in this step receives as input the output of the corresponding note classifier, from the classification stage, as well as, the output for some preceding and following frames, and gives as output a *yes* or *no* answer. This way, it assesses whether the middle frame of the sequence contains the specific note or not. See Fig. 4, below, for an example.

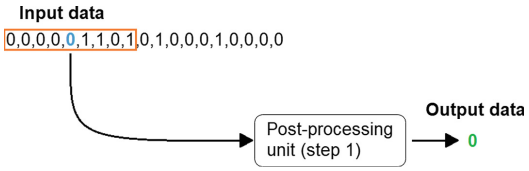

Fig. 4. Example of how a post-processing unit from step 1 works. The 1's represent a frame that was identified with a specific musical note and the 0's the opposite scenario. The squared window around a portion of the input data represents the sequence given to the post-processing unit. The number in the middle of that window, represents the frame that the unit is trying to predict. Finally, the number represented below the *output data* illustrates the actual prediction from the post-processing unit.

Note that during the post-processing, the squared window represented in the figure above, will slide to the right, one frame at a time, until the last frame, and for each sequence contained on that window, an output prediction is given for the middle frame. When finished, the whole set of new predictions represents the resultant transcription of the system (see Fig. 5).

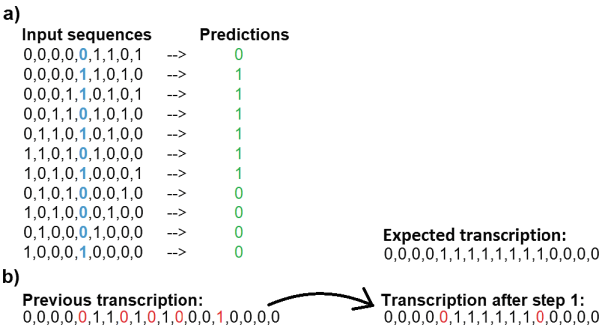

Fig. 5. Representation of all the sequences given to step 1, regarding the previous example. The numbers in red represent wrongly transcribed frames. (a) All the input sequences given to the unit and its resultant predictions. (b) Representation of the previous and new transcription.

It is important to point out that, the sequences given to a post-processing unit do not contain binary data (only zeros or ones) but, instead, values between 0 and 1 (because the classifiers' output unit use the sigmoid activation function). However, for ease of understanding all the examples given in this section represent those values as binary data.

Step 2 - Fix notes duration according to onsets

For further improvement, an additional post-processing step was created (see Fig. 6). This new step is like the previous one since it receives as input a sequence of previously transcribed frames from step 1 and it also tries to predict the possible transcription for the middle frame of that sequence. However, it also receives two additional sequences: one sequence with the original transcription from the classification stage and another one based on the output of an onset detector algorithm [19]. An onset consists in the start time of a musical note.

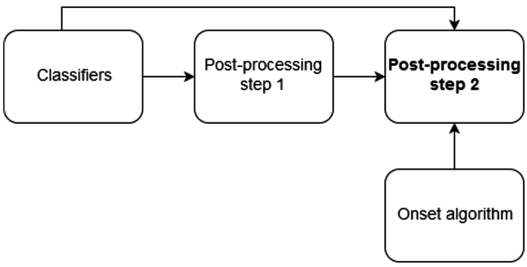

Fig. 6. Representation of the three different sequences received by step 2 post-processing units.

The rationale behind the concept of receiving the original transcription from the classification stage is based on *stacked* systems [20], where an additional system receives as input the output of the previous step, as well as, the original input.

Note that the onset detector algorithm applied is not perfect and is also not able to distinguish between onsets of different musical notes. Thus, these post-processing units need to deal with problems like: (i) falsely and missing detected onsets and (ii) onsets of other musical notes.

Step 3 - Fix notes onset

To refine our model in terms of onset detection, an additional post-processing step was added. In this step, only the frames predicted as note onsets are targeted. Specifically, for each predicted note onset, these post-processing units decide whether a readjustment is needed or not. Therefore, they can output three possible transformations: *SHIFT LEFT*, *ACCEPT* and *SHIFT RIGHT*. An example of the three possible transformations is shown in Fig. 7.

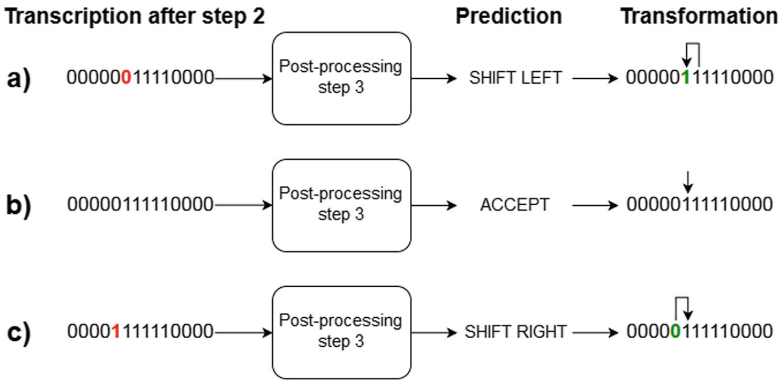

Fig. 7. Three possible transformations in step 3 of the post-processing stage.

This post-processing unit receives two sequences as input. One with the corresponding transcription of the note onset and nearby frames (previous and following four frames), and a second sequence, with the output of the onset algorithm, used in the previous step. Thus, regarding the example represented in Fig. 7, the input data received by this post-processing unit could be as demonstrated in Fig. 8:

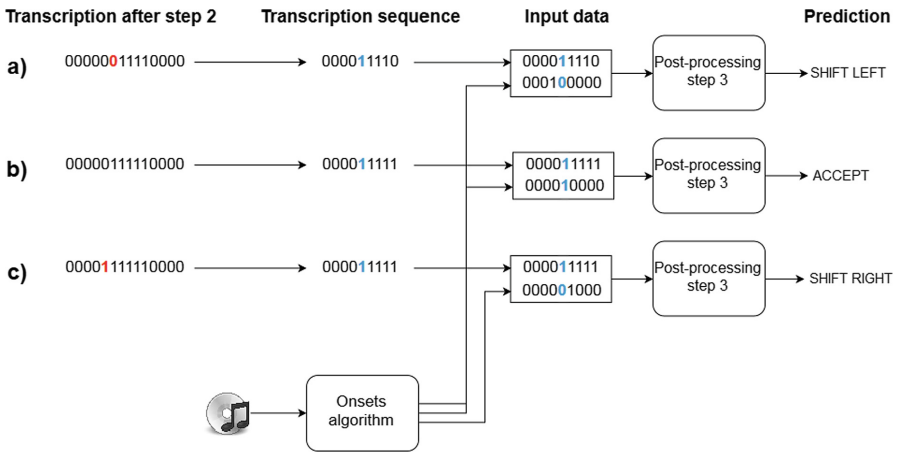

Fig. 8. Illustration of the given input data with all three types of possible transformations. (a) Scenario where the note onset should be readjusted to one frame before. (b) Scenario where the onset is considered already correct. (c) Scenario where the onset should be shifted to one frame after.

4 Results

In this section, results and a comparison with similar techniques used in other state-of-the-art works are presented. First, the dataset is described, followed by the metrics used for comparing our approach. Then, the results obtained are shown, and finally, a comparison with other research works is given.

4.1 Dataset

To be able to compare our approach with already existent ones, we use the *Configuration 1* dataset from [2], based on MAPS [21]. This dataset comprises four folds, each one containing a different combination of musical pieces, with 216 musical pieces in the training set and 54 pieces in the testing set. This means that, for each fold, a transcription system comprised of 88 ANNs for the classification stage and 264 ANNs (88×3) for the post-processing stage, need to be created.

4.2 Metrics

We use both frame-based and note-based metrics [22] to compare our model. Frame-based metrics consists on evaluating frame-by-frame the final transcription, whereas note-based consists on evaluating each transcribed musical note by considering its pitch and its onset. Regarding the note onset, we also assume a tolerance of $\pm$ 50 ms.

We use *precision*, *recall* and *f-measure* for both frame-based and note-based evaluation metrics. Mathematically, these metrics can be expressed as:

$$Precision\,(P) = \frac{TP}{TP + FP} \tag{1}$$

$$Recall\,(R) = \frac{TP}{TP + FN} \tag{2}$$

$$F\text{-}measure\,(F) = \frac{2 \times P \times R}{P + R}, \tag{3}$$

where TP represents true positives, which consist on correctly identified frames/notes, FP represents false positives, which consist on wrongly detected frames/notes and FN represents false negatives, which consist on missed detected frames/notes.

4.3 Results and Comparison

The obtained results from our model, per each step are presented in Table 1, below.

Table 1. Obtained results per each step.

Stage/step	Frame-based (F)	Note-based (F)
Classifiers	66.89	33.08
Post-processing step 1 - Fix notes duration	79.78 (+12.89)	51.61 (+18.53)
Post-processing step 2 - Fix notes duration according to onsets	80.23 (+0.45)	55.97 (+4.36)
Post-processing step 3 - Fix onsets	80.47 (+0.24)	61.64 (+5.67)

From the table above, we may conclude that the post-processing stage played an essential role in the improvement of the transcription results. The frame-based metrics were improved by an amount of 13.58% and the note-based by 28.56%. To better evaluate how distant our system is from the expected transcription, a portion of the resultant and expected transcription from the musical piece *BMW 846 Prelude in C Major* from J. S. Bach is shown in Fig. 9.

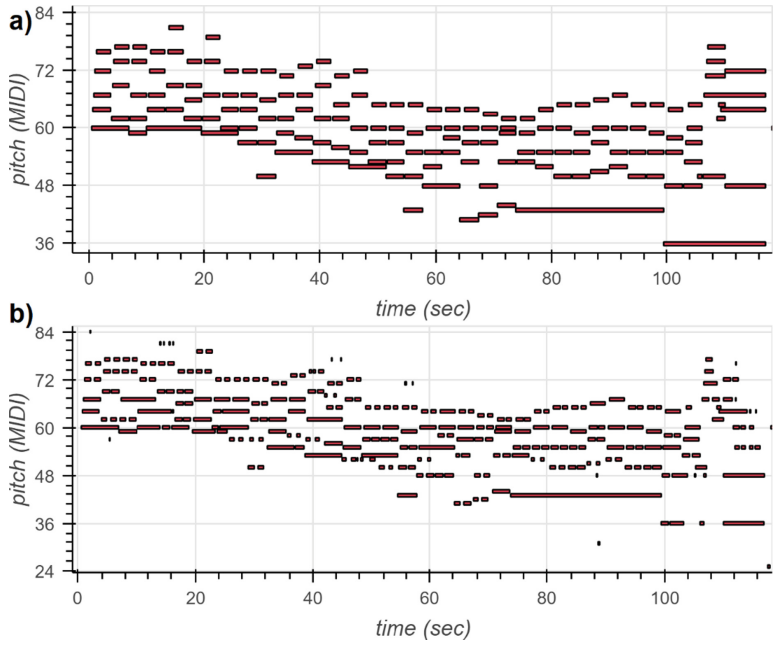

Fig. 9. Portion of the expected and resulting transcription of the *BMW 847 Prelude in C Major*. (a) Expected transcription. (b) Resulting transcription.

To compare our results, two state-of-the-art works were chosen: [7] and [8]. Both apply the same dataset as well as the same type of ANNs. The comparison is shown in Table 2.

Table 2. Comparison with two other state-of-the-art works.

Approach	Frame-based			Note-based		
	P	*R*	*F*	*P*	*R*	*F*
ANN [7]	65.66	70.34	67.92	**62.62**	**63.75**	**63.179**
ANN [8]	76.63	70.12	73.11	–	–	–
Ours	**84.94**	**76.46**	**80.47**	61.81	61.54	61.64

From the table above, one can conclude that our approach significantly surpasses both works in frame-based metrics, while reaching similar results in note-based.

To assess our approach against more recent artificial intelligence techniques such as Recurrent Neural Networks (RNNs) or Convolutional Neural Networks (CNNs), we further compare it to three other systems, taking advantage of the fact that these three systems have also used the same dataset, thus making the comparison feasible. Table 3 encompasses the results from all the approaches.

Table 3. Comparison with works that apply more recent types of artificial neural networks.

Approach	Frame-based			Note-based		
	P	*R*	*F*	*P*	*R*	*F*
RNN [7]	67.89	70.66	69.25	64.64	65.85	65.24
CNN [7]	72.45	76.56	74.45	**67.75**	**66.36**	**67.05**
CNN [8]	80.19	**78.66**	79.33	–	–	–
Ours	**84.94**	76.46	**80.47**	61.81	61.54	61.64

Even when our approach is compared with works that implement more recent types of ANNs, it still yields higher frame-based metrics than those systems, and, at the same time, it reaches comparable results in note-based metrics. This demonstrates the effectiveness of our approach.

5 Conclusions and Future Work

In this paper, we tackled the AMT problem using a *divide and conquer* approach. The obtained results show that this is a promising path for tackling the AMT problem, since they surpassed current state-of-the-art works in frame-based metrics and reach similar metrics in note-based, even when compared with other systems that apply more recent types of artificial neural networks. The use of artificial neural networks as post-processing units demonstrated to be essential for improving the whole performance of the system. In the future, a comparison could be done between post-processing units that take advantage of artificial neural networks and traditional statistical methods, such as Hidden Markov Models, in order to understand which one is better.

To conclude, there is still plenty of space for future work. For instance, in the case of the classifiers, other techniques could be used, like Recurrent Neural Networks or Convolutional Neural Networks. In addition, an improved version of the onset algorithm could also be used. Thus, a possible solution could be the creation of an additional ANN to filter false positives from this original onset algorithm, or, instead, create an onset algorithm from-scratch using deep learning techniques, like some authors propose [23].

References

1. Schmidhuber, J.: Deep learning in neural networks: an overview. Neural Netw. **61**, 85–117 (2015)
2. Sigtia, S., Benetos, E., Dixon, S.: An end-to-end neural network for polyphonic piano music transcription. IEEE/ACM Trans. Audio Speech Lang. Process. **24**(5), 927–939 (2016)
3. Kelz, D., Korzeniowski, B., Arzt, W.: On the potential of simple framewise approaches to piano transcription. In: 17th International Society for Music Information Retrieval Conference (2016)
4. Hawthorne, C., et al.: Onsets and frames: dual-objective piano transcription. arXiv preprint arXiv:1710.11153 (2017)
5. Marolt, M.: A connectionist approach to automatic transcription of polyphonic piano music. IEEE Trans. Multimed. **6**, 439–449 (2004)
6. Inácio, T., Miragaia, R., Reis, G., Grilo, C., Fernandéz, F.: Cartesian genetic programming applied to pitch estimation of piano notes. In: 2016 IEEE Symposium Series on Computational Intelligence, pp. 1–7 (2016)
7. Moorer, J.A.: On the segmentation and analysis of continuous musical sound by digital computer (1975)
8. Lea, A.P.: Auditory modeling of vowel perception. Ph.D. thesis, University of Nottingham, United Kingdom (1992)
9. Bello, J.P., Sandler, M.: Blackboard system and top-down processing for the transcription of simple polyphonic music. In: Proceedings of the COST G-6 Conference on Digital Audio Effects, pp. 7–9 (2000)
10. Klapuri, A.P.: Multiple fundamental frequency estimation based on harmonicity and spectral smoothness. IEEE Trans. Speech Audio Process. **11**(6), 804–816 (2003)
11. Smaragdis, P., Brown, J.C.: Non-negative matrix factorization for polyphonic music transcription. In: 2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pp. 177–180 (2003)
12. Emiya, V., Badeau, R., David, B.: Multipitch estimation of quasi-harmonic sounds in colored noise. In: 10th International Conference on Digital Audio Effects (2007)
13. Yeh, C.: Multiple fundamental frequency estimation of polyphonic recordings. Ph.D. thesis, University Paris, France (2008)
14. Reis, G.M.J.D.: Una aproximación genética a la transcripción automática de música. Ph.D. thesis, University of Extremadura, Spain (2012)
15. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
16. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)

17. Neelakantan, A., et al.: Adding gradient noise improves learning for very deep networks. arXiv preprint arXiv:1511.06807 (2015)
18. Montavon, G., Orr, Geneviève B., Müller, K.-R. (eds.): Neural Networks: Tricks of the Trade. LNCS, vol. 7700. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35289-8
19. Martins, L.G.P.M.: A computational framework for sound segregation in music signals. Ph. D. thesis, University of Porto, Portugal (2008)
20. Zhang, H., et al.: StackGAN: text to photo-realistic image synthesis with stacked generative adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5907–5915 (2017)
21. Emiya, V., Badeau, R., David, B.: Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle. IEEE Trans. Audio Speech Lang. Process. **18**(6), 1643–1654 (2010)
22. Bay, M., Ehmann, A.F., Downie, J.S.: Evaluation of multiple-F0 estimation and tracking systems. In: The International Society of Music Information Retrieval, pp. 315–320 (2009)
23. Eyben, F., Böck, S., Schuller, B., Graves, A.: Universal onset detection with bidirectional long-short term memory neural networks. In: Proceedings 11th International Society for Music Information Retrieval Conference, pp. 589–594 (2010)