PLOS ONE

RESEARCH ARTICLE

Robust Real-Time Music Transcription with a Compositional Hierarchical Model

Matevž Pesek[1]*, **Aleš Leonardis**[1,2], **Matija Marolt**[1]

1 University of Ljubljana, Faculty of Computer and Information Science, Laboratory for computer graphics and multimedia, Ljubljana, Slovenia, **2** University of Birmingham, School of Computer Science, Centre for Computational Neuroscience and Cognitive Robotics, Birmingham, United Kingdom of Great Britain and Northern Ireland

* matevz.pesek@fri.uni-lj.si

Citation: Pesek M, Leonardis A, Marolt M (2017) Robust Real-Time Music Transcription with a Compositional Hierarchical Model. PLoS ONE 12 (1): e0169411. doi:10.1371/journal.pone.0169411

Editor: Constantine Dovrolis, Georgia Institute of Technology, UNITED STATES

Received: July 11, 2016

Accepted: December 17, 2016

Published: January 3, 2017

Data Availability Statement: All audio files (which were compiled and annotated by the authors) are available from the URL (http://musiclab.si/folkmusic.zip), and we also uploaded the data to the Open Science Framework repository as one of the suggested repositories by PLOS ONE. The reference and link are as follows: Pesek, M. (2016, December 21). Folk Song Dataset. Retrieved from osf.io/f7h3r.

Funding: The author(s) received no specific funding for this work.

Competing Interests: The authors have declared that no competing interests exist.

Abstract

The paper presents a new compositional hierarchical model for robust music transcription. Its main features are unsupervised learning of a hierarchical representation of input data, transparency, which enables insights into the learned representation, as well as robustness and speed which make it suitable for real-world and real-time use. The model consists of multiple layers, each composed of a number of parts. The hierarchical nature of the model corresponds well to hierarchical structures in music. The parts in lower layers correspond to low-level concepts (e.g. tone partials), while the parts in higher layers combine lower-level representations into more complex concepts (tones, chords). The layers are learned in an unsupervised manner from music signals. Parts in each layer are compositions of parts from previous layers based on statistical co-occurrences as the driving force of the learning process. In the paper, we present the model's structure and compare it to other hierarchical approaches in the field of music information retrieval. We evaluate the model's performance for the multiple fundamental frequency estimation. Finally, we elaborate on extensions of the model towards other music information retrieval tasks.

Introduction

Music information retrieval (MIR) deals with extraction of semantic descriptions from music in its various forms. As in many related areas, a significant increase in algorithm accuracy and efficiency has been achieved in recent years for tasks such as melody estimation [1, 2], chord estimation [3–5], beat tracking [6, 7], mood [8] and genre estimation [9, 10], and pattern analysis [11–13].

Recently, parallel to other areas, deep learning has been successfully introduced to the MIR [14, 15]. A deep learning algorithm constructs multiple levels of data abstraction (a hierarchy of features) in order to model high-level representations present in the observed data [16]. Several deep learning models have been applied to different MIR tasks, such as deep neural networks, convolutional neural networks (CNNs) and deep belief networks (DBNs). One of the first uses of deep architectures for analyzing audio signals was presented by Lee [17], who applied convolutional DBNs for speaker identification. Later, Hamel and

Eck [18], evaluated DBNs for genre recognition using a five-layer DBN with three hidden layers for feature extraction. Since then, deep architectures achieved promising results on a variety of tasks: Schmidt and Kim [19] used a five-layer DBN for extraction of emotion-based acoustic features, Pikrakis [20] showed that DBNs can be used for rhythm genre discrimination, conditional DBNs were used by Battenberg and Wessel [21] for drum pattern analysis, while Schmidt [22] showed that DBNs can be trained to understand rhythm and melody. Other architectures have also been used, for example recurrent neural networks for audio chord estimation [5, 23] and convolutional neural networks for key detection [24] and onset detection [25, 26].

The goal of music transcription is to estimate a music score (notes played) from an audio signal. Its essential part is the multiple fundamental frequency estimation, where the goal is to estimate all the fundamental frequencies (corresponding to pitches) in individual time-frames of a music signal. As an important MIR goal, transcription has been researched since the early 1970s and a variety of approaches have been developed [27–30]. Some approaches use note hypothesis evaluation based on the signal spectrum [31, 32], while others [33–35] model the audio signal as a composition of sources. Several approaches are tuned to the transcription of specific instruments [36–39] or focus on transcribing instrument-specific symbolic data [40].

Several neural-network-based deep approaches were also presented for music transcription [41–44]. Bock and Schedl [41] used a recurrent neural network model for a piano transcription, while Nam et al. [42] combined deep belief networks with support vector machines and a hidden Markov model for the same task. Rigaud and Radenen [44] proposed a combination of two deep neural networks for transcription of singing voice.

However, music transcription approaches are rarely evaluated on the real-world recordings, which may not have been recorded in ideal studio environments or with professional performers. This is in large part due to the lack of diverse annotated datasets currently available—most datasets consist mainly of the synthesized recordings, which are easily obtainable, and contain only a small number of annotated real recordings. Consequently, the robustness of the algorithms may suffer, as they may overfit the small datasets and the instrument timbres, which leads to poor performance on diverse materials and in the presence of noise.

This paper introduces a novel compositional hierarchical model for the multiple fundamental frequency estimation (MFFE). The proposed model can be regarded as a novel deep architecture with unsupervised learning and a transparent structure, which allows for representation and interpretation of the signal's content on different levels of complexity. The model's main feature is the relativity of learned concepts, which enables construction of compact and robust models. The main contribution of this paper is a model which can perform robustly on datasets that vary in audio and source quality, with real-time computation and affordable spatial requirements. This makes the model useful for a wide range of applications and with music recordings of varying quality.

The presented model is an extension of the model first introduced in [45] for three different MIR tasks: chord estimation, mood estimation and MFFE. Its structure and learning algorithm are improved, resulting in higher MFFE accuracy. Additionally, the experimental part is significantly extended in this paper using four datasets for a cross-dataset evaluation. We also adapted the model for melodic pattern extraction in the symbolic domain [46], where we evaluated it on the JKU PDD dataset. For each musical piece, the model was built independently and inferred with the same piece. Patterns were represented by activations of parts on the top layers.

The paper is structured as follows: the proposed model is described in the first Section. The evaluation of the model is provided in the second Section, followed by discussion. The last Section concludes the paper and gives ideas for future work.

Compositional hierarchical model for MIR

The main principle of compositional hierarchical models lies in the hierarchical nature of our perception of the world. Just like our visual system can discern complex forms by combining basic elements like edges, lines, contrasts and colors into increasingly more complex percepts, so can our auditory system group frequency components into auditory events, multiple tonal events into harmonies, their time evolution into melodies and harmonic progressions.

Hierarchical music representations are intuitive when considering the spectral and temporal structures in music. The generative theory of tonal music [47] may well be the first examples of hierarchical music modeling in musicology. Although the model itself mostly relies on expert rules, the hierarchical structuring is a good fit, since it is based on patterns of human perception and cognitive processes. Other attempts [48, 49] have been made to empirically evaluate such hierarchical representations produced by human cognitive processes. The approaches based on temporal hierarchical structures [50] have been presented, taking human short-term memory into consideration, while defining a rule-based model for auditory processing. Hierarchical models also abound in analysis of music perception from the point of view of computational biology and neuroscience [51–54].

We propose a compositional hierarchical model designed specifically for music signal processing. The model can learn a hierarchical representation of audio signals in an unsupervised manner, starting from signal components on the lowest layer, up to individual music events on the highest layers.

The structure of our model is inspired by the research in the field of computer vision, specifically the *learned Hierarchy of Parts* (lHoP) model presented by Leonardis and Fidler [55, 56]. Their model represents objects in images in a hierarchical manner, structured in layers, from simple to complex image parts. The model is learned from the statistics of natural images and can be employed as a robust statistical engine for object categorization and other computer vision tasks.

We show that a similar approach can also be used for music representation and analysis. Our model is built on the assumption that a complex signal can be decomposed into a hierarchy of building blocks—*parts*. The parts exist at various levels of granularity and represent sets of entities describing the signal. According to their complexity, parts can be structured across layers from the less to the more complex. The parts on higher layers are expressed as compositions of parts on lower layers, analogous to the fact that a chord is composed of several pitches, and each pitch of several harmonic partials. A part can therefore describe individual frequencies in a signal, their combinations, as well as pitches, chords and temporal patterns, such as chord progressions. The entire structure is *transparent*, so that the role of each part can be observed and interpreted.

The presented model differs from the aforementioned lHoP model in its concept. While it shares the inspiration for its hierarchical composition of structures and statistical learning, the CHM was developed from scratch with focus on MIR tasks. The input to the CHM is a spectral audio representation, which significantly influences its structure. Consequently, the mechanisms for activations, part compositions and layers were redefined to meet the specifics of such representation. Inhibition and hallucination mechanisms, also inspired by the lHoP model, were newly defined according to the new model structure. Additionally, an automatic gain control mechanism that incorporates time dimension into CHM processing was newly introduced specifically for this model.

Model structure

The compositional hierarchical model consists of the input layer $\mathcal{L}_0$ and several compositional layers $\{\mathcal{L}_1, \ldots, \mathcal{L}_N\}$. Each compositional layer $\mathcal{L}_n$ contains a set of parts $\{P_1^n, \ldots, P_M^n\}$, where a part is a composition of parts from $\mathcal{L}_{n-1}$ and may itself be part of any number of compositions on $\mathcal{L}_{n+1}$. Thus, the compositional model forms a hierarchy of parts, as may be observed in Fig 1, where connections between the parts represent the structure of compositions.

Compositional layers. Layers $\{\mathcal{L}_1, \ldots, \mathcal{L}_N\}$ contain parts which are compositions of parts from lower layers. Formally, we define the composition P_i^n as:

$$P_i^n = \{P_{k_0}^{n-1}, \{P_{k_j}^{n-1}, (\mu_j, \sigma_j)\}_{j=1}^{K-1}\}. \tag{1}$$

P_i^n is a composition of K parts from layer $\mathcal{L}_{n-1}$—*subparts*. The composition is governed by the parameters $\mu_{1,\ldots,K-1}$ and $\sigma_{1,\ldots,K-1}$ which model relations between subparts. These relations are *relative*, meaning that the compositions are defined by the relative distances (*offsets*) between the subpart $P_{k_0}^{n-1}$ and the subparts $P_{k_1}^{n-1}, \ldots, P_{k_{K-1}}^{n-1}$. The offsets are encoded by parameters $\mu_{1,\ldots,K-1}$ and $\sigma_{1,\ldots,K-1}$ and always defined relative to $P_{k_0}^{n-1}$ which we denote as the composition's *central* part. For example, P_2^2 in Fig 1 is defined as:

$$P_2^2 = \{P_1^1, \{P_3^1, (1200, 25)\}\}, \tag{2}$$

where μ and σ are given in cents. It represents a composition of P_1^1 with P_3^1 spaced approximately 1200 cents (one octave) apart, where σ governs the allowed deviation from this value. Since all relationships in the model are relatively encoded, rather than encoding specific instances of a music concept (e.g. the tone A5), our model learns generalized concepts (e.g. a tone is a set of frequency components at some relative positions). The benefits of such relative encoding are discussed in the *Relativity and shareability of parts* Section. All compositions and their parameters are learnt in an unsupervised manner, as explained in the *Learning* Section.

The mapping from relatively defined to absolutely positioned concepts (e.g. a generalized tone concept to the tone A5) is performed during an *inference* on an input audio signal, by calculating part *activations* upwards through all the layers (see *Inference* Section).

A part *activation* indicates that the concept it represents was found in the input signal. An activation has two components: a *location*, which maps the part onto the frequency axis, thus making it absolute, and a *magnitude*, representing its strength. A part can activate only if all of its subparts are activated with magnitude greater than zero (this constraint can be relaxed as described in the *Inference* Section). Due to the relative encoding of the concepts in the model, a part can simultaneously activate at multiple locations, indicating that the concept it represents was found at several locations in the input signal.

The activation location of part P_i^n at time t is defined as:

$$A_L^{(t)}(P_i^n) = A_L^{(t)}(P_{k_0}^{n-1}). \tag{3}$$

Thus, central parts of compositions propagate their locations upwards through the hierarchy. With respect to the example in Eq 2, when P_1^1 is activated at $440Hz$, P_3^1 at $880Hz$, and P_2^2 is activated at 440Hz. Such propagation of the locations through the central parts represents a very useful indexing mechanism, which enables an efficient top-down analysis of part activations from the upper to the lower layers, adding to the transparency of the model.

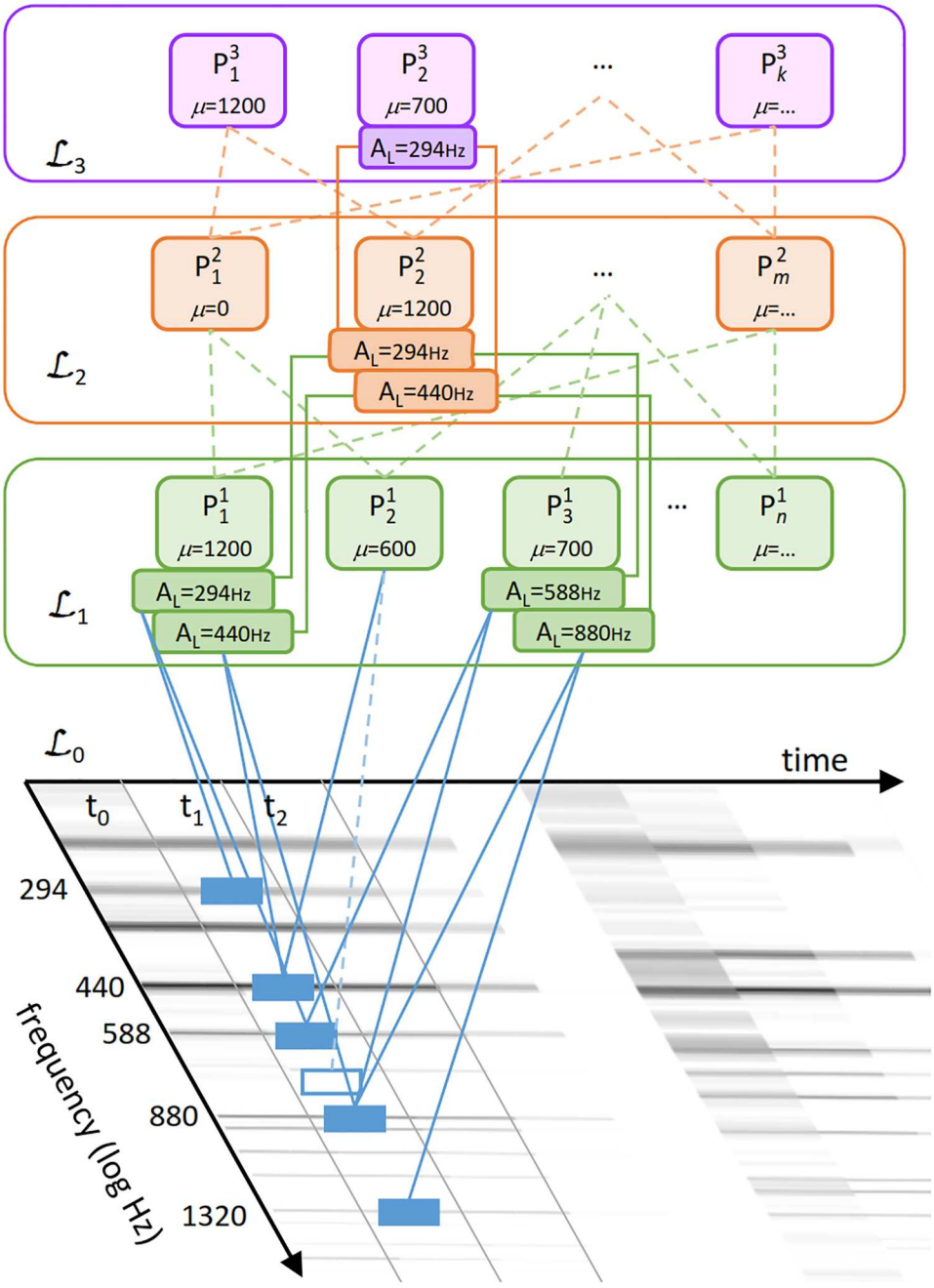

Fig 1. The compositional hierarchical model. The input layer corresponds to the signal components in the time-frequency representation. The parts on higher layers are compositions of the lower-layer parts (depicted as connections between parts, parameter μ is given in cents). A part may be contained in several compositions, e.g. P_1^1 is a part of compositions P_1^2, P_2^2 and P_m^2. Active parts have activation locations displayed underneath, a part can have several activations on different locations. The entire structure is transparent, thus we can discern that the activation of P_2^2 at 294*Hz* represents a tone (harmonic series) starting at 294 Hz by observing the subtree leading from the activation to $\mathcal{L}_0$.

doi:10.1371/journal.pone.0169411.g001

The activation magnitude is defined as a weighted sum of subpart magnitudes:

$$w_j = \begin{cases} 1: & j = 0 \\ \mathcal{N}(A_L^{(t)}(P_{k_0}^{n-1}) - A_L^{(t)}(P_{k_j}^{n-1}), \mu_j, \sigma_j): & j > 0 \end{cases}, \tag{4}$$

$$A_M^{(t)}(P_i^n) = \tanh\left(\frac{1}{K}\sum_{j=0}^{K-1} w_j A_M^{(t)}(P_{k_j}^{n-1})\right)$$

where the weights w_i are defined by the match between the locations of the subpart activations and the composition parameters μ and σ.

Input layer. The input layer $\mathcal{L}_0$ models a time-frequency representation of the input signal X. It consists of the single atomic part P_1^0, which is activated at locations of all the frequency components in the signal at a given time-frame t. Thus, for any frequency bin k, P_1^0 is activated as:

$$\begin{aligned} A_L^{(t)}(P_1^0) &= f(k) \\ A_M^{(t)}(P_1^0) &= |X_t(k)|, \end{aligned} \tag{5}$$

where $f(k)$ represents the frequency of the frequency bin k and $|X_t(k)|$ its magnitude.

Relativity and shareability of parts

The proposed model has two important features that set it apart from similar architectures.

The *relativity* of parts enables a single part to represent an abstract high-level concept regardless of its location in the input signal. Relative perception naturally occurs in human learning process. It is an important part of the abstraction of the object of interest, and enables the formation of a complete percept, regardless of its environment. It minimizes the amount of memory needed to store the learned concepts and enables their robust identification in previously unobserved sensory inputs, such as within noisy audio signals and in the presence of non-musical events.

Relativity is inherent in our model and can be observed in the definitions of part composition and activation (Eqs 1 and 4). Although the parts are relative and only represent abstract concepts with no direct absolute representation (e.g. the model cannot encode the pitch G5 explicitly, but only the concept of pitch), the part's activation at a given location indicates where and when a given concept appears in the signal. Since this can occur at several locations, a part can have multiple activations at different locations. This is also shown in Fig 1, where P_1^1, P_3^1 and P_2^2 have two activations each, meaning that the concepts they represent are present at several locations in the signal.

The relative nature of parts that enables the representation of concepts regardless of their location also enables efficient *shareability* of the parts. A single part on the layer $\mathcal{L}_{n-1}$ may be a part of several compositions on the layer $\mathcal{L}_n$. Consequently, any two or more $\mathcal{L}_{n-1}$ parts may form a number of different $\mathcal{L}_n$ compositions at different offsets. Thus, they may be combined into several more complex abstractions, themselves relative.

The consequence of relativity and shareability is that the model can very efficiently encode complex concepts. As an example: a part representing the concept of pitch may be shared by several compositions on a higher level that encode different intervals. This encoding is general, compact and efficient if we consider the alternative of encoding all the intervals in an absolute manner. This is also evident in the evaluation of the proposed model (see *Evaluation* Section), where a learned hierarchy with a small number of compositions is shown to be robust and to

generalize well in modeling musical events in audio signals, which differ from the ones used for training in quality, the amount of noise and the number and the type of sources present in the signal.

Learning

The model is constructed layer-by-layer with unsupervised learning on a set of input signals, starting with $\mathcal{L}_1$. We view the learning as an optimization problem, where we aim to find a minimal set of compositions for the learned layer, which will explain the maximal amount of information present in the input data. The learning process is driven by the statistics of part activations which capture regularities in the input data.

To formalize the problem, we first define the *coverage* of a part's activation at the time t as a set of $\mathcal{L}_0$ activations (spectral components) which have caused the activation. This set can be obtained efficiently by observing the tree formed by the activated subparts through indexing encoded in the locations of their central parts down to the layer $\mathcal{L}_0$ as:

$$\begin{aligned} A_C^{(t)}(P_i^n) &= \bigcup_{j=0}^{K-1} A_C^{(t)}(P_{k_j}^{n-1}) \\ A_C^{(t)}(P_1^0) &= \{k : f(k) \in A_L^{(t)}(P_1^0)\}. \end{aligned} \tag{6}$$

The coverage of the entire layer $\mathcal{L}_n$ is the set of spectral components in the input data, which all the parts in the layer cover:

$$A_C^{(t)}(\mathcal{L}_n) = \bigcup_{p \in \mathcal{L}_n} A_C^{(t)}(p) \tag{7}$$

The goal of learning a new layer $\mathcal{L}_n$ is to minimize the amount of uncovered information in the input data and, on the other hand, to limit the number of parts added to the layer, which can be expressed as:

$$min(\sum_t \sum_{k \notin A_C^{(t)}(\mathcal{L}_n)} |X_t(k)|^2 + \lambda |\mathcal{L}_n|), \tag{8}$$

where λ is a regularization factor which balances between the number of parts and the adequacy of the coverage.

The problem of finding an optimal coverage is a special case of the well-known set cover problem, which is NP-complete. We therefore approximate the solution by using a greedy algorithm, which incrementally adds compositions to the new layer. With each iteration the algorithm chooses a composition that covers the largest amount of uncovered data. The entire learning algorithm is composed of two steps: finding new candidate compositions and adding compositions to a new layer.

Finding candidate compositions. When learning the layer $\mathcal{L}_n$, we first need to form a set of new compositions, which will be considered for inclusion in the new layer. We perform inference on the training set up to the layer $\mathcal{L}_{n-1}$, and then observe the co-occurrences of $\mathcal{L}_{n-1}$ part activations over the entire training set. The co-occurrences provide information on the parts, which frequently activate simultaneously and are thus believed to form a common concept. We calculate the histograms of co-occurring activations according to distances between activation locations. New compositions are formed from parts where the number of co-occurrences exceeds a learning threshold τ_L. The composition parameters μ and σ are estimated from the corresponding histogram (Fig 2) and each new composition is added to the set of candidate compositions $\mathcal{C}$.

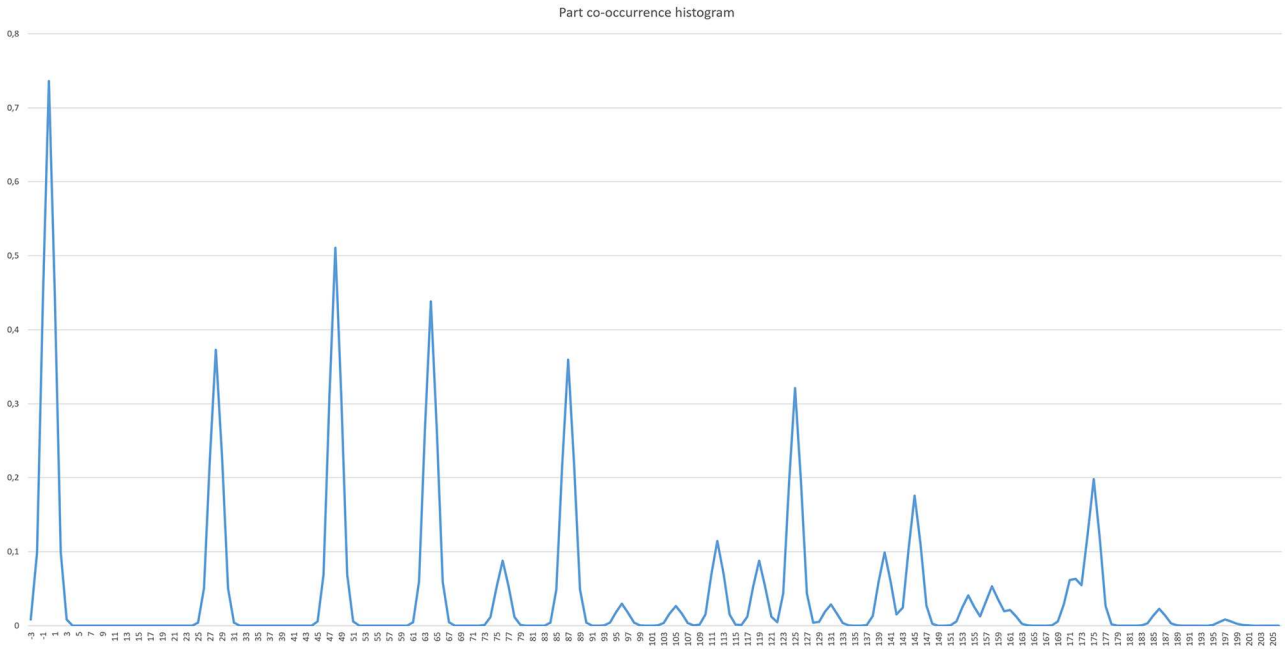

Fig 2. Co-occurrence histogram for an $\mathcal{L}_2$ part. The normalized co-occurrence histogram represents the distribution of distances (offsets) of $\mathcal{L}_2$ subparts that activate simultaneously. The distances are shown relative to a chosen central $\mathcal{L}_2$ part.

doi:10.1371/journal.pone.0169411.g002

Selecting compositions. Due to the NP-completeness of the set cover problem, we use a greedy approach to select a subset of compositions from the set $\mathcal{C}$, which leaves a minimal amount of information in the training set uncovered (according to Eq 8). In each iteration, a composition from $\mathcal{C}$, which contributes the most to the coverage of the training set, is selected and added to the new layer. This ensures that only compositions which provide enough new information with regard to the currently selected set will be added. The selection is stopped when either: a sufficient percentage of information in the learning set is covered (according to the threshold τ_P), or no part from the candidate set adds enough to the coverage of information (according to τ_C). The algorithm in Fig 3 outlines the described approach.

The learning proceeds layer-by-layer, starting at $\mathcal{L}_1$, until the complexity of the layer parts achieves the desired complexity of modeled musical events, depending on the underlying problem. The chosen values of thresholds and their impact on training are described in the *Evaluation* Section.

Inference

Inference is the process of calculating part activations on an input signal according to Eqs 3 and 4. Inference is calculated bottom-up and layer-by-layer, whereby the time-frequency representation of the input signal serves as the input of the layer $\mathcal{L}_0$. The observed locations and magnitudes of activations yield insight into the analyzed signal through the concepts that the activated parts represent and can be used as features for further processing.

In this Section, we describe three additional mechanisms that can be used during an inference to increase the predictive power and robustness of the model: *hallucination*, *inhibition* and *automatic gain control*.

Hallucination. When calculating activations, the default model behavior is very conservative—a part is activated only if all of its subparts are activated (*Model structure* Section). *Hallucination* relaxes this condition and enables the model to produce activations even in the case of

1: procedure SELECT($\mathcal{C}$)
2: $prevCov \leftarrow 0$
3: $cov \leftarrow \emptyset$
4: $\mathcal{L}_n \leftarrow \emptyset$
5: $sumEnergy \leftarrow \sum_t \sum_k |X_t(k)|^2$
6: **repeat**
7: **for** $P \in \mathcal{C}$ **do**
8: $c \leftarrow 0$
9: **for all** t **do**
10: $\mathcal{F} \leftarrow A_C^{(t)}(\mathcal{L}_n \cup P)$
11: $c \leftarrow c + \sum_{k \in \mathcal{F}} |X_t(k)|^2$
12: **end for**
13: $cov[P] \leftarrow c/sumEnergy$
14: **end for**
15: $Chosen \leftarrow \underset{P}{\operatorname{argmax}}(cov)$
16: $\mathcal{L}_n \leftarrow \mathcal{L}_n \cup Chosen$
17: $\mathcal{C} \leftarrow \mathcal{C} \setminus Chosen$
18: **if** $cov[Chosen] - prevCov < \tau_C$ **then**
19: **break**
20: **end if**
21: $prevCov \leftarrow cov[Chosen]$
22: **until** $prevCov > \tau_P \vee \mathcal{C} = \emptyset$
23: **return** $\mathcal{L}_n$

Fig 3. Greedy algorithm for the selection of compositions from the candidate set $\mathcal{C}$. Compositions that add the most to the coverage of information in the learning set are prioritized.

doi:10.1371/journal.pone.0169411.g003

incomplete (missing, masked or damaged) input. The model generates activations of parts, which most fittingly cover the information, present in the input signal, where fragments, which are not present, are "hallucinated". The missing information is thus extrapolated from the knowledge acquired during learning, encoded into the model structure.

Hallucination changes the conditions under which a part may be activated. It is governed by the parameter τ_H, which can be defined per layer. By hallucination, the part P_i^n is activated when the percentage of positive spectral components it covers exceeds the τ_H:

$$\frac{|\{k : k \in A_C^{(t)}(P_i^n) \wedge |X_t(k)| > 0\}|}{|A_C^{(t)}(P_i^n)|} \geq \tau_H. \tag{9}$$

If we set τ_H to 1, we obtain the default behavior (all of the covered spectral components must be present in the signal for parts to activate), while lowering of the parameter value leads to an increased number of activations across all layers.

By allowing activations in the presence of incomplete input, hallucination not only enables the model to fill-in the missing information, but also to yield the alternative explanations of the input signal. Namely, different parts of the model can explain the same fragments of information in the input. Hallucination boosts these alternative representations and enables the model to produce multiple explanations of the same input.

Inhibition. Inhibition performs the hypothesis refinement by reducing the number of part activations on individual layers. It provides a balancing factor in the model by reducing redundant activations, similar to lateral inhibition in the human auditory system [57]. Although the learning algorithm penalises parts redundantly covering the signal, some redundant parts are always present. During inference, each layer may therefore produce multiple redundant activations covering the same information in the input signal (hallucination also adds to the number of such activations).

The activation of the part P_i^n is inhibited when different parts on the same layer cover the same spectral components in the input signal, but with a higher activation magnitude:

$$\exists\{P_j^n..P_k^n\} : \wedge \begin{cases} \dfrac{|A_C^{(t)}(P_i^n) \setminus \bigcup\{A_C^{(t)}(P_j^n)..A_C^{(t)}(P_k^n)\}|}{|A_C^{(t)}(P_i^n)|} < \tau_I \\ \forall A_M^{(t)}(P_{j..k}^n) > A_M^{(t)}(P_i^n) \end{cases}, \tag{10}$$

where τ_I controls the amount of inhibition. Such control is needed, as complete inhibition of redundant parts' activations is undesirable, due to the robustness the activations provide in a form of competing hypotheses about the information in the input signal. For example, a value of 0.5 will cause an activation to be inhibited if half of its coverage is already covered by stronger activations of other parts.

Alongside the hypothesis refinement, the removal of redundant activations also reduces noise in the input signal, which is usually manifested in a number of low-magnitude activations of parts on various layers. In combination with hallucination, inhibition provides an efficient way to control the explanatory power and robustness of the proposed model.

Automatic gain control. The model presented so far is time-independent. It operates on a time-frame-by-time-frame basis, where each time-frame in the time-frequency representation is processed independently from others. The automatic gain control mechanism (AGC) was introduced in the inference process in order to model short-time dependencies between frames. It operates on principles similar to automatic gain control contrast mechanism in human [58] and animal [59] perceptual systems. The mechanism allows linking of part activations through time by introducing time dependencies between activations.

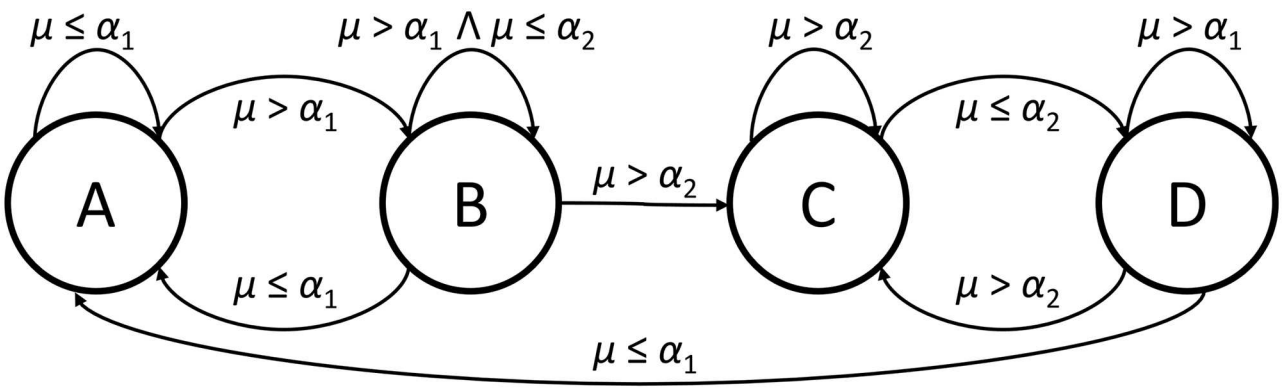

Fig 4. A finite state machine implementing the AGC mechanism. State A represents the normal behavior of a part, state B the boosting (onset), state C the sustain and state D the decay of the activation magnitude.

doi:10.1371/journal.pone.0169411.g004

The operation of the AGC is defined with a four-state finite state machine, as shown in Fig 4. AGC changes the activation of a part in the following manner: when the part is activated at a new location, and its activation persists, activation magnitude is initially boosted to accentuate the onset and later suppressed towards a stable value (see Fig 5).

The four AGC states represent: (A) normal part behavior, (B) onset, (C) sustain and (D) decay state. Transitions between the states are conditioned on the density of part activations θ within the time window W, which for the part P_i^n at the time t is defined as:

$$\theta = \frac{1}{W} \| [A_M^{(t-W+1)}(P_i^n), \ldots, A_M^{(t)}(P_i^n)] \|_0. \quad (11)$$

α_1 and α_2 are thresholds that control transitions between the states. The magnitude of a part

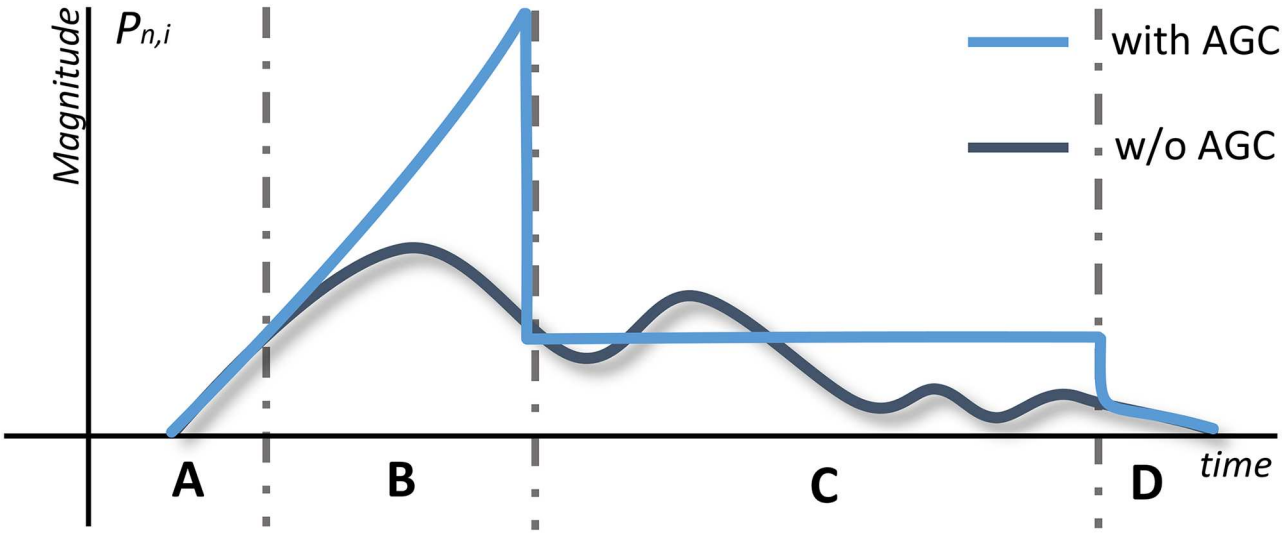

Fig 5. An abstract representation of AGC influence on part activations. Without AGC, activation magnitudes may notably fluctuate, especially towards the end of an event. AGC boosts the onset of an event and later keeps the activation magnitude on a fixed level until the offset.

doi:10.1371/journal.pone.0169411.g005

activation for the individual states is calculated as:

$$A_M^{(t)}(P_i^n) = \begin{cases} A_M^{(t)}(P_i^n): & A, D \\ \sum_{f=t-W+1}^{t} A_M^{(f)}(P_i^n): & B \\ \tau_S: & C \end{cases}, \quad (12)$$

where τ_S represents a constant activation magnitude in the sustain state.

The mechanism operates on all layers; it has a short-term effect on lower layers and longer-term effect on higher layers (the window size W increases for each consecutive layer) in line with the complexity of concepts represented on different layers. The mechanism's effect on the activation magnitude is shown in Fig 5. AGC stabilizes activations, boosts event onsets and produces an overall smoother model output with less fluctuation.

Evaluation

Our proposed model is applicable to different MIR tasks in the audio domain, as presented in [45], as well as in the symbolic domain [46]. In this section, we demonstrate its usefulness for the multiple fundamental frequency estimation (MFFE), where the goal is to estimate which fundamental frequencies are present in the signal at individual time-frames.

Choice of parameters

Our model has several parameters, which influence learning and inference. We first evaluated the sensitivity of the proposed model to different values of its two most significant parameters: τ_H and τ_I. Results in Fig 6 show that the model performance for MFFE is mostly stable, apart from extreme values. If τ_H that controls hallucination is set to a low value, the amount of activations increases drastically, as parts are allowed to hallucinate almost freely and vice-versa. High values produce few activations, so in both cases performance suffers. Similarly, a low value of τ_I (inhibition) results in a large number of part activations and subsequently worse performance.

Other parameters also have well defined roles and effects. The model is invariant to changes of τ_P above approximately 0.75 due to limitations imposed by τ_C. High values of the latter result in small part candidate sets and insufficient coverage of the signal. AGC parameters α_1 and α_2 influence the stability of activations over time and only affect the performance if set to extreme values.

Because the model is not very sensitive to values of its parameters, we did not tune them specifically for each experiment, but chose to set them to common-sense values and keep them constant for all experiments. The input layer $\mathcal{L}_0$ was based on a constant-Q transform with 345 frequency bins between 55 and 8000 Hz (48 per octave), a step size of 10 ms and a maximal window size of 100 ms. Training and the inference parameters τ_H, τ_I, τ_P and τ_C were set to values 0.7, 0.5, 0.9 and 0.005 respectively and AGC parameters to $\alpha_1 = 0.2$ and $\alpha_2 = 0.5$.

Experiment

To evaluate the model for the multiple fundamental frequency estimation, we trained three layers of compositions on top of $\mathcal{L}_0$, as described in the previous Section. A four-layer structure was sufficient for the model to learn a robust representation of pitch, as shown in our results.

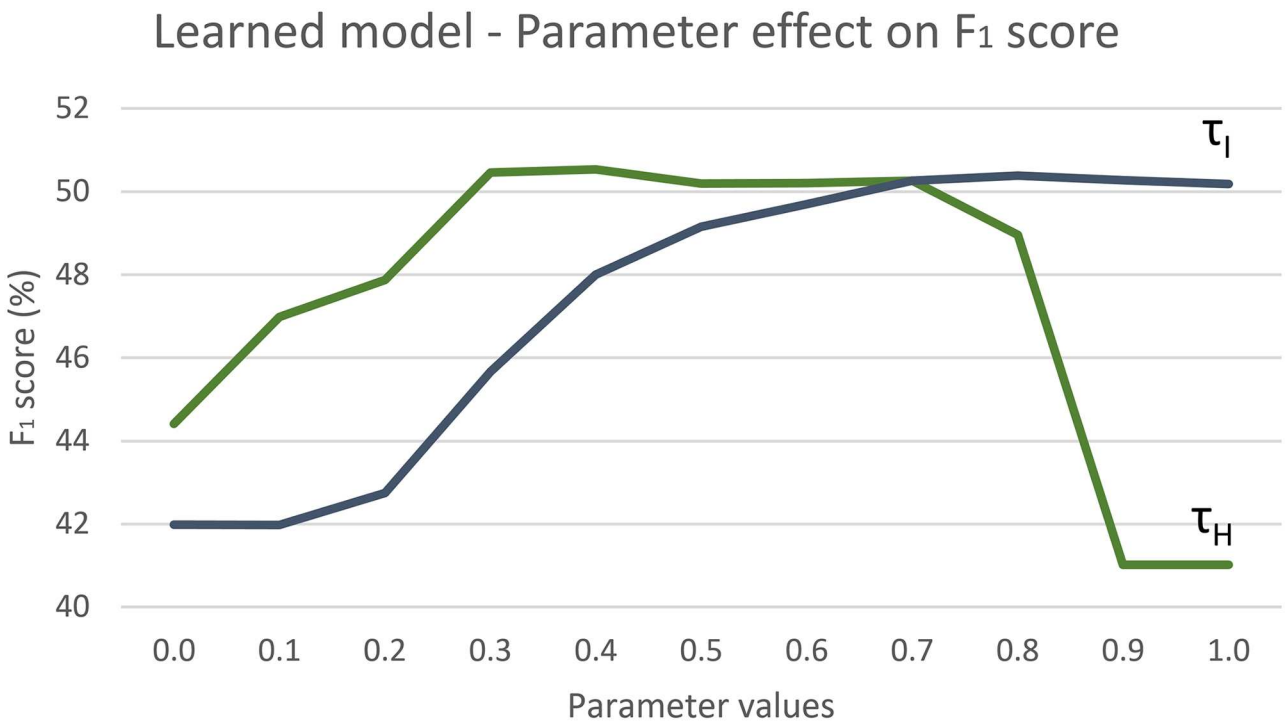

Fig 6. Piano transcription performance on the *AkPnBcht* folder from the MAPS dataset with different τ_H and τ_I values. The *x* axis represents parameter values, the *y* axis the F_1 score.

doi:10.1371/journal.pone.0169411.g006

Training is performed in an unsupervised manner on a training dataset. To assess how different training datasets influence the structure of the model, we trained the model on several large and small datasets: three small datasets consisting of individual isolated instrument sounds (piano, flute and guitar), two medium-sized datasets of popular music (the Beatles and Queen albums) and a large dataset of polyphonic piano music. A comparison of the learned structures showed that the size of the learned models did not vary significantly. All models contained a small number of compositions on all layers (in total between 50–60), with very similar structures. The average Jaccard index per layer was 0.586 and 0.381 for $\mathcal{L}_1$ and $\mathcal{L}_2$ respectively. It was higher for models trained only on individual instrument samples (0.764 and 0.56) or only polyphonic music (0.778 and 0.522). Only identical parts were counted when calculating the index, although other parts were also similar (e.g. compositions that have three out of four subparts and offsets in common). Such small size and similarity of the learned models is the consequence of two features of the proposed model: relativity and shareability, which enable learning of generalized concepts, such as pitch, encoded in small models, which can be trained on small datasets.

We therefore decided to perform all our experiments on a model trained on individual Bösendorfer model 225 (From the EastWest Ultimate Piano Collection) piano notes (these were not included in the testing datasets), which makes training fast, but still yields good results. The learned model contained only 23, 12 and 16 parts on layers 1, 2 and 3 respectively. The amount of parts on $\mathcal{L}_3$ layer did not exceed those on $\mathcal{L}_1$ and $\mathcal{L}_2$, as could be expected in compositional models, because regularization during the learning process balances the coverage and the amount of generated parts per layer.

To use the model for MFFE, we exploited its transparency, which enabled us to interpret the activations of parts on the layer $\mathcal{L}_3$ and directly map them onto the frequency axis, thus

Table 1. Comparison of CHM, DNMF, Klapuri and Benetos approaches. F_1 scores in %, running times and memory usage for 1 minute of audio for different datasets and different transcription methods are shown. F_1 scores are frame-based scores calculated in accordance to MIREX MFFE evaluations [63].

Dataset	CHM	DNMF	Klapuri [29]	Benetos [27]	Benetos [62]
MAPS MIDI	52.6	**61.6**	56.0	56.7	56.7
MAPS D	51.8	57.1	52.5	50.1	**62.6**
Su & Yang	**48.9**	32.6	48.0	40.3	55.6
Folk song	**49.3**	35.0	31.8	27.5	16.2
Average F_1	**50.7**	46.6	47.1	43.7	47.8
Running time (s)	6.2	5.7	19.4	188.1	87
RAM Usage (MB)	63.8	120.0	43.2	1914.2	716.5

doi:10.1371/journal.pone.0169411.t001

extracting a set of fundamental frequencies at each time-frame. No additional supervised machine learning models were therefore used for estimation of fundamental frequencies.

To assess the robustness of the learned pitch concepts, we tested the model for MFFE on four distinct datasets: MAPS M [60], containing piano-synthesized MIDI files, MAPS D, containing recordings of the Disklavier [60], Su & Yang dataset [61], containing mixtures of piano and string instruments, and a dataset of folk songs sung by choirs of 2–4 singers (available at osf.io/f7h3r).

For all datasets, we compared our results to three other methods: DNMF decomposition of the time-frequency representation [38], where DNMF was trained on 70% of the dataset and tested on the remaining 30%, the Klapuri's multiple F0 estimation method [29] and two approaches presented by Benetos and Weyde [27, 62]. For the Klapuri's method, we used 30% of the annotated dataset to fine-tune the salience threshold parameter.

Results are given in Table 1. They show that the proposed model learns a robust representation of pitch and has good generalisation power, as it yields consistent results on different datasets. While other approaches, such as DNMF or Benetos', achieve better scores on some datasets, they overfit the timbres they were trained on (e.g. DNMF was trained on the majority of the MAPS dataset), so their performances in cases where timbre is not so well defined (e.g. the folk song dataset containing choir singing) are poor.

Although trained only on piano notes, the proposed model unsupervisedly learned the concept of pitch in a robust manner, without (over)fitting to specific templates of a single instrument. It is the most accurate of all compared approaches on the folk song dataset, where it demonstrates its robustness. A singing transcription is difficult for most algorithms based on harmonic templates (which include all compared algorithms), as the vocal timbre changes not only between songs (different performers), but also within a song (different vowels, stress etc.). It is therefore difficult to capture the timbre with a template, which results in poor transcription performance, especially in terms of precision. In addition, these songs originate from field recordings of folk music that are performed by amateur singers and recorded in everyday environments with portable audio equipment. Thus, they significantly differ from the studio-level or synthesised recordings. The CHM, with its multilayer representation, hallucination, inhibition and AGC mechanisms, achieves performance comparable to other datasets, while the compared methods perform significantly worse (Kruskal-Wallis test $\chi^2 = 56.8$, $p < 10^{-11}$).

Error analysis

We analysed the model's output with respect to the manually annotated ground truth to assess the most typical errors made by the proposed model. Four types of errors are frequent: offset localisation, semitone errors, harmonic (octave) errors and pitch fluctuation.

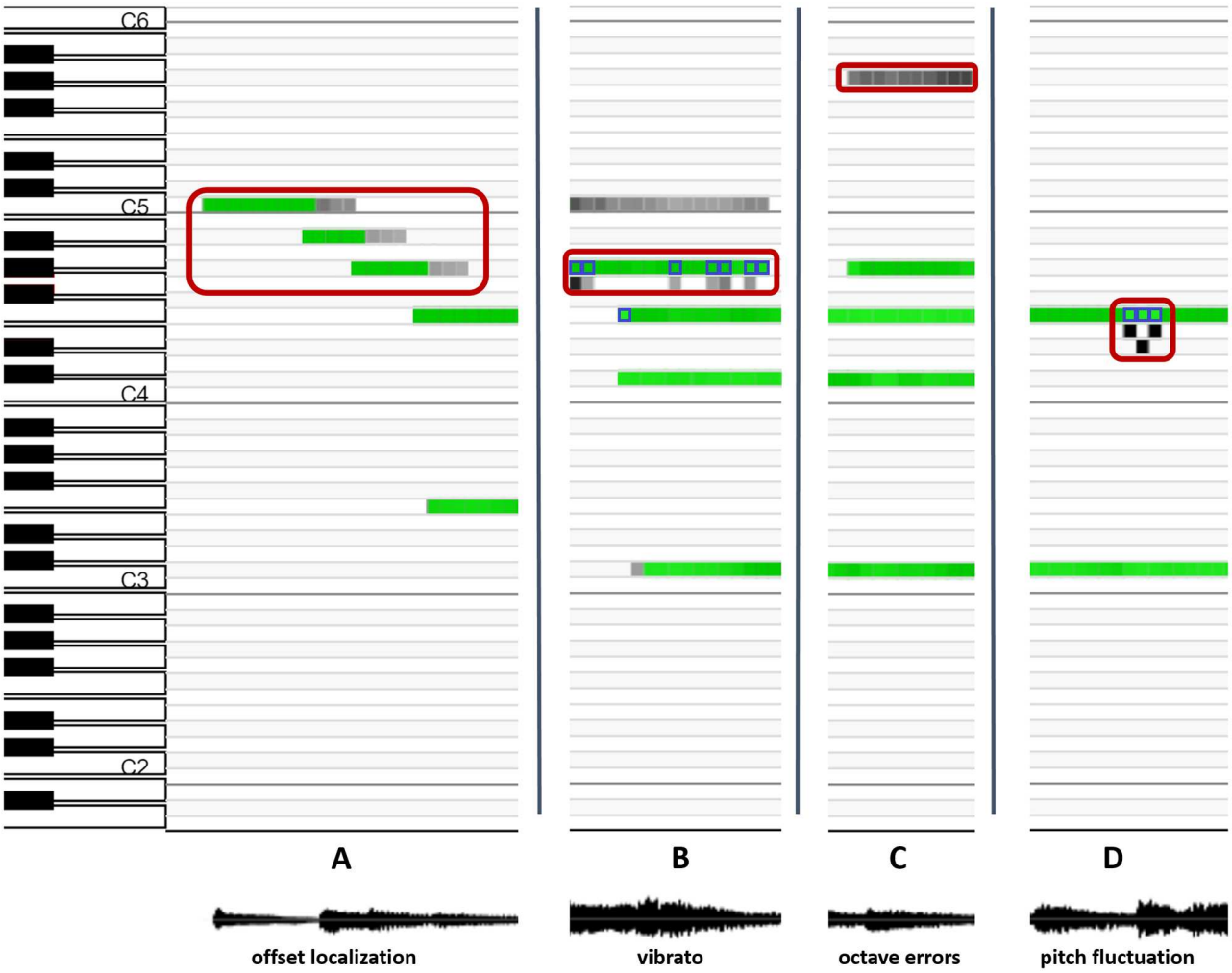

Fig 7. The most frequent errors produced by the CHM. Ground truth annotations are displayed in green, the CHM activations are shown in grey. Activations that are not aligned with the ground truth represent false positive errors. Additionally, false negatives are outlined with blue color.

doi:10.1371/journal.pone.0169411.g007

Offset localisation errors frequently appear in recordings with strong reverberation, where an event is prolonged and is detected after the instrument stopped playing. The AGC mechanism may additionally prolong the detected offsets, so the combination of both factors reflects in longer durations of identified events, as shown in Fig 7-A. The singer's vibrato can cause the detected pitch to shift up or down in individual frames, which may cause semitone errors, as the groundtruth usually reflects the desired and not the actual pitch within a time-frame (Fig 7-B). Octave and other harmonically related errors are a common source of errors for most algorithms due to sharing of harmonics between harmonically related tones. CHM is no exception, especially in recordings where instruments contain many strong harmonic components (Fig 7-C). Voice fluctuations are commonly present in singing, especially when singers sing a capella (without a support of instruments). Pitch may fluctuate at onsets of syllables, resulting in the spread of energy over several semitones, similar to vibrato, which leads to pitch estimates that differ from the ground truth, as may be observed in Fig 7-D.

Discussion

The proposed CHM model offers a novel approach to music information retrieval and analysis, specifically for the music transcription. It provides a compact hierarchical representation of the content of a music signal through activations of learned concepts over several layers. We demonstrated its effectiveness by using a model learned in a completely unsupervised manner for the multiple fundamental frequency estimation. This was possible due to the model's transparency, where part activations can be interpreted meaningfully and projected to the input layer.

When compared to specialized approaches, the proposed algorithm may not perform as well as the current state of the art, which is expected, as it is not tuned for a specific task. For comparison, one of the best MAPS M transcription scores is 77.1%, reported by Weninger et al. [37]. His approach differs significantly from ours—it is based on a support vector machine classifier, which was trained on a large portion of the MAPS dataset (approx. 80% of the dataset).

The deep network approaches for MFFE [41–44] also typically use a large proportion of the dataset for training. Bock and Schedl [41] evaluated a recurrent neural network model on four piano music datasets, including MAPS MIDI and MAPS D. They reported a high F_1 score (up to 93.5%) for note onset detection; however, they also used a significant amount of the datasets for training and validation (approximately 75% and 9.4% on average per dataset for training and validation respectively). Nam et al. [42] reported results for 30 second excerpts from the MAPS dataset (74.4% frame-level F_1 score) by using roughly 60% of the dataset for training and 25% for validation.

The reason for this large proportions of training samples is that MFFE datasets are relatively small. This is due to the fact that annotations require expert knowledge and a significant amount of time. The annotations can thus not be crowdsourced, as for example in image labelling, where deep networks are very successful. It thus becomes necessary to include a significant amount of the available data into a training set, retaining only a small portion (down to 10% in several cases) for testing. Results are assumed to generalize over the whole dataset, and there is no information on how these models would perform on more diverse datasets, and for instruments with different timbres.

In comparison, our model was trained on only a small set of 88 piano key samples not present in the MAPS dataset. Although the CHM does not reach the accuracy of such tuned approaches, it is able to generalize and perform well in a variety of cases where the source is not so well defined, as shown in our evaluation on the Su & Yang and Folk song datasets. We may therefore conclude that the CHM extracts timbre-invariant features from the audio signal, which, combined with a robust inference mechanism, lead to a stable performance in various scenarios.

Real-time performance

An added feature of the proposed approach lies in the small sizes of learned models, which are consequences of part relativity and shareability. The computational complexity of inference with such small models is low, so CHM can be used for transcription in real-time scenarios. Table 1 lists running times and memory consumption of all compared algorithms for one minute of audio measured on a system with 16GB RAM and an Intel Xeon E5520 2.26GHz processor using a single thread. The CHM and DNMF are the fastest, with approximately ten-to-one ratio of audio length over processing time, followed by Klapuri (approx. three-to-one ratio). Both approaches by Benetos and Weyde with 1.5-to-three and one-to-three ratio are not usable in real-time scenarios, as next to high running times, they also require the entire audio file for

processing. The memory consumption of the proposed approach is also low—it uses approximately half the memory in comparison to DNMF, and around 50% more than Klapuri's approach.

In addition, the approach is parallelizable, as parts on a layer can be inferred independently and thus in parallel. The speed, small memory consumption and robustness of our approach make it suitable for real-world use, and applicable within embedded systems and mobile devices with multiple cores and low processing power per core.

Conclusion and future work

We introduced a compositional hierarchical model for music information retrieval and music analysis. We showed how the model is used for music transcription and evaluated its ability to perform in real-time. This ability enables the usage of the model for a number of real-world applications and platforms, such as embedded and mobile systems. The model is constructed by unsupervised learning on a set of audio recordings and contains compositions of parts reflecting the statistical regularities in the learning set, encoding simple concepts on lower layers and complex concepts on higher layers. The model's transparency enables insights into the model's structure and consequently into the music concepts represented by individual parts, such as pitch partials, pitches and harmonies. The relativity and shareability of parts enable a compact representation of the learned concepts, while robustness is achieved by incorporating inhibition, hallucination and AGC mechanisms, as presented in the *Inference* Section.

A different deep architecture

The compositional hierarchical model shares some similarities with deep learning architectures. It is similar in terms of learning a variety of signal abstractions on several layers of granularity. The learning procedure is similar: the structure is built layer-by-layer. However, unlike most deep architectures, CHM is learned in an entirely unsupervised manner, so no annotated dataset is needed for training and validation. In addition, several aspects of the model set it apart from other architectures.

Transparency is manifested in the compositional nature of the model. Parts are compositions of subparts and their activations are directly observable and interpretable (each activation can be projected to the input layer and its effect observed). In contrast, most other neural-network-based deep architectures offer no clear explanation of the underlying feature extraction process and the meaning of the extracted features, with the exception of convolutional neural networks, which partially and indirectly offer explanations of their nodes [64]. Transparency enables the model to be used directly as a classifier by observing and interpreting part activations, as we show in our evaluation task.

In addition, the hallucination and the inhibition mechanisms facilitate the production of alternative explanations of the input during inference. By suppressing the most prominent explanation, the model can produce alternative hypotheses previously suppressed by this explanation. Combined with transparency, this makes the model a suitable music analysis tool, where signal contents can be visualized and interpreted as activations of high-to-low level concepts encoded by the model, which may also be interactively manipulated to obtain alternative hypotheses.

Relativity and shareability of parts enable efficient encoding of the learned concepts and lead to a small number of parts needed to represent complex concepts. A part in the proposed model is defined by the relative distance between its subparts and can be activated on different locations along the frequency axis. Therefore, the large amount of layer units that, for example, convolutional networks need to cover the entire range of frequencies, is not necessary.

Relativity is accompanied by part shareability: parts on a layer may be shared by many compositions on higher layers. Although this feature is similar to other deep representations, relativity takes shareability a step further: a set of subparts may form several new relative compositions on a higher layer representing different entities and may thus be efficiently reused. The learned models therefore contain a small number of parts, which also enables the use of small datasets for training a small number of trainable parameters, which lead to a very fast inference. This is also evident in the presented evaluation, where a small set of samples was used to train a three-layer model that performed well on several different datasets.

Future work

The model is general and can be used for audio-based as well as symbolic MIR tasks, including automated chord estimation and mood estimation [45], symbolic pattern discovery [46] and multiple fundamental frequency estimation presented in this paper. For the latter, the model serves both as a feature extractor, as well as a classifier. We also demonstrated the model's robustness to varying timbres and audio signals which were recorded in suboptimal conditions.

The proposed approach is naturally expandable to the time domain, which we already demonstrated by its application to the symbolic pattern discovery task [46]. In our future work, we aim to further develop the model as a general purpose model for music information retrieval and music analysis. Future work includes stacking of models applied to the audio and symbolic domains, thus introducing a single model which covers different MIR tasks and is suitable for real-world application. We intend to extend the model to encode long-term temporal dependencies of music events, thus encoding concepts such as melodic lines, chord progressions and rhythmic patterns. Such a unified framework which models the spatial (frequency) and temporal structure of music events should improve performance for a variety of MIR tasks and potentially eliminate the need for additional temporal processing stages, such as the hidden Markov models.

Author Contributions

Conceptualization: MP MM AL.

Data curation: MP MM.

Formal analysis: MP MM.

Investigation: MP.

Methodology: MM.

Project administration: MM AL.

Resources: MM.

Software: MP MM.

Supervision: MM AL.

Validation: MM.

Visualization: MP MM.

Writing – original draft: MP MM AL.

Writing – review & editing: MP MM AL.

References

1. Ryynänen MP, Klapuri AP. Automatic Transcription of Melody, Bass Line, and Chords in Polyphonic Music. Computer Music Journal. 2008; 32(3):72–86. doi: 10.1162/comj.2008.32.3.72
2. Bittner RM, Justin S, Essid S, Bello JP. Melody Extraction By Contour Classification. In: Proceedings of the International Conference on Music Information Retrieval (ISMIR). Malaga; 2015. p. 500–506.
3. Harte C, Sandler M, Abdallah S, Gomez E. Symbolic representation of musical chords: A proposed syntax for text annotations. In: Proceedings of the International Conference on Music Information Retrieval (ISMIR). London; 2005.
4. Papadopoulos H, Peeters G. Large-case Study of Chord Estimation Algorithms Based on Chroma Representation and HMM. Content-Based Multimedia Indexing. 2007;53–60.
5. Sigtia S, Boulanger-Lewandowski N, Dixon S. Audio Chord Recognition With A Hybrid Recurrent Neural Network. In: Proceedings of the International Conference on Music Information Retrieval (ISMIR). Malaga; 2015. p. 127–133.
6. Holzapfel A, Davies MEP, Zapata JR, Oliveira JL, Gouyon F. Selective Sampling for Beat Tracking Evaluation. IEEE Transactions on Audio, Speech, and Language Processing. 2012; 20(9):2539–2548. doi: 10.1109/TASL.2012.2205244
7. Durand S, Bello JP, David B, Richard G. No Title. In: Acoustics, Speech and Signal Processing (ICASSP); 2015. p. 409–413.
8. Laurier C, Meyers O, Serrà J, Blech M, Herrera P, Serra X. Indexing music by mood: design and integration of an automatic content-based annotator. Multimedia Tools and Applications. 2009; 48(1):161–184. doi: 10.1007/s11042-009-0360-2
9. Tzanetakis G, Cook P. Musical genre classification of audio signals. IEEE Transactions on Speech and Audio Processing. 2002; 10(5):293–302. doi: 10.1109/TSA.2002.800560
10. Anglade A, Ramirez R, Dixon S. Genre Classification Using Harmony Rules Induced from Automatic Chord Transcriptions. In: Proceedings of the International Conference on Music Information Retrieval (ISMIR). Kobe; 2009. p. 669–764.
11. Conklin D. Discovery of distinctive patterns in music. Intelligent Data Analysis. 2010; 14(5):547–554.
12. Meredith D, Lemstrom K, Wiggins GA. Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. Journal of New Music Research. 2002; 31(4):321–345. doi: 10.1076/jnmr.31.4.321.14162
13. Wang Ci, Hsu J, Dubnov S. Music Pattern Discovery with Variable Markov Oracle: A Unified Approach to Symbolic and Audio Representations. In: Proceedings of the International Conference on Music Information Retrieval (ISMIR)2. Malaga; 2015. p. 176–182.
14. Humphrey EJ, Bello JP, LeCun Y. Moving beyond feature design: deep architectures and automatic feature learning in music informatics. In: Proceedings of the International Conference on Music Information Retrieval (ISMIR). Porto; 2012.
15. Humphrey EJ, Cho T, Bello JP. Learning a Robust Tonnetz-Space Transform for Automatic Chord recognition. In: Acoustics, Speech and Signal Processing (ICASSP). New York; 2012. p. 453–456.
16. Bengio Y, Courville A, Vincent P. Representation learning: a review and new perspectives. IEEE transactions on pattern analysis and machine intelligence. 2013; 35(8):1798–828. doi: 10.1109/TPAMI.2013.50 PMID: 23787338
17. Lee H, Pham P, Largman Y, Ng AY. Unsupervised feature learning for audio classification using convolutional deep belief networks. In: Advances in Neural Information Processing Systems; 2009. p. 1096–1104.
18. Hamel P, Eck D. Learning Features from Music Audio with Deep Belief Networks. In: Proceedings of the International Conference on Music Information Retrieval (ISMIR); 2010. p. 339–344.
19. Schmidt EM, Kim YE. Learning emotion-based acoustic features with deep belief networks. In: 2011 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA). IEEE; 2011. p. 65–68.
20. Pikrakis A. A Deep Learning Approach to Rhythm Modelling with Applications. In: 6th International Workshop on Machine Learning and Music, held in conjunction with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD 2013; 2013. p. 1–4.
21. Battenberg E, Wessel D. Analyzing Drum Patterns using Conditional Deep Belief Networks. In: Proceedings of the International Conference on Music Information Retrieval (ISMIR); 2012. p. 37–42.
22. Schmidt EM, Kim YE. Learning Rhythm and Melody Features with Deep Belief Networks. In: Proceedings of the International Conference on Music Information Retrieval (ISMIR); 2013. p. 21–26.

23. Boulanger-Lewandowski N, Bengio Y, Vincent P. Audio chord recognition with recurrent neural networks. In: Proceedings of the International Conference on Music Information Retrieval (ISMIR); 2013.
24. Dieleman S, Brakel P, Schrauwen B. Audio-based Music Classification with a Pretrained Convolutional Network. In: Proceedings of the International Conference on Music Information Retrieval (ISMIR). Miami; 2011. p. 24–28.
25. Schluter J, Bock S. Musical Onset Detection with Convolutional Neural Networks. In: 6th International Workshop on Machine Learning and Music, held in conjunction with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD 2013; 2013.
26. Ullrich K, Schluter J, Grill T. Boundary detection in music structure analysis using convolutional neural networks. In: Proceedings of the International Conference on Music Information Retrieval (ISMIR). Taipei; 2014. p. 417–422.
27. Benetos E, Weyde T. Multiple-F0 estimation and note tracking for Mirex 2015 using a sound state-based spectrogram factorization model. In: 11th Annual Music Information Retrieval eXchange (MIREX'15). Malaga; 2015. p. 1–2.
28. Gerhard D. Pitch Extraction and Fundamental Frequency: History and Current Techniques. Regina: University of Regina, Saskatchewan, Canada; 2003.
29. Klapuri A, Davy M, editors. Signal Processing Methods for Music Transcription. New York: Springer; 2006.
30. Klapuri AP. Automatic Music Transcription as We Know it Today. Journal of New Music Research. 2004; 33(3):269–282. doi: 10.1080/0929821042000317840
31. Roebel A, Rodet X. Multiple Fundamental Frequency Estimation and Polyphony Inference of Polyphonic Music Signals. IEEE Transactions on Audio, Speech, and Language Processing. 2010; 18 (6):1116–1126. doi: 10.1109/TASL.2009.2030006
32. Pertusa A, Iñesta JM. Efficient methods for joint estimation of multiple fundamental frequencies in music signals. EURASIP Journal on Advances in Signal Processing. 2012; 2012(1):27. doi: 10.1186/1687-6180-2012-27
33. Dessein A, Cont A, Lemaitre G. Real-time polyphonic music transcription with non-negative matrix factorization and beta-divergence. In: Proceedings of the International Conference on Music Information Retrieval (ISMIR); 2010. p. 489–494.
34. Grindlay G, Ellis DPW. Transcribing Multi-Instrument Polyphonic Music With Hierarchical Eigeninstruments. IEEE Journal of Selected Topics in Signal Processing. 2011; 5(6):1159–1169. doi: 10.1109/JSTSP.2011.2162395
35. Smaragdis P, Brown JC. Non-negative matrix factorization for polyphonic music transcription. In: 2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (IEEE Cat. No.03TH8684). IEEE; 2003. p. 177–180.
36. Marolt M. A Connectionist Approach to Automatic Transcription of Polyphonic Piano Music. IEEE Transactions on Multimedia. 2004; 6(3):439–449. doi: 10.1109/TMM.2004.827507
37. Weninger F, Kirst C, Schuller B, Bungartz HJ. A discriminative approach to polyphonic piano note transcription using supervised non-negative matrix factorization. In: Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP). Vancouver; 2013. p. 6–10.
38. Boulanger-Lewandowski N, Bengio Y, Vincent P. Discriminative Non-Negative Matrix Factorization For Multiple Pitch Estimation. In: Proceedings of the International Conference on Music Information Retrieval (ISMIR). Porto, Portugal; 2012. p. 205–210.
39. Vincent E, Bertin N, Badeau R. Adaptive Harmonic Spectral Decomposition for Multiple Pitch Estimation. IEEE Transactions on Audio, Speech, and Language Processing. 2010; 18(3):528–537. doi: 10.1109/TASL.2009.2034186
40. Barbancho AM, Klapuri A, Tardon LJ, Barbancho I. Automatic Transcription of Guitar Chords and Fingering From Audio. IEEE Transactions on Audio, Speech, and Language Processing. 2012; 20(3):915–921. doi: 10.1109/TASL.2011.2174227
41. Bock S, Schedl M. Polyphonic Piano Note Transcription with Recurrent Neural Networks. In: Proceedings of the 37th International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2012. p. 121–124.
42. Nam J, Ngiam J, Lee H, Slaney M. A Classification-Based Polyphonic Piano Transcription Approach Using Learned Feature Representations. In: Proceedings of the International Conference on Music Information Retrieval (ISMIR). Miami; 2011. p. 175–180.
43. Kelz R, Dorfer M, Korzeniowski F, Bock S, Arzt A, Widmer G. On the Potential of Simple Framewise Approaches to Piano Transcription. In: Proceedings of the International Conference on Music Information Retrieval (ISMIR)2. New York; 2016. p. 475–481.

44. Rigaud F, Radenen M. Singing Voice Melody Transcription using Deep Neural Networks. In: Proceedings of the International Conference on Music Information Retrieval (ISMIR). New York; 2016. p. 737–743.
45. Pesek M, Leonardis A, Marolt M. A compositional hierarchical model for music information retrieval. In: Proceedings of the International Conference on Music Information Retrieval (ISMIR). Taipei; 2014. p. 131–136.
46. Pesek M, Medvešek U, Leonardis A, Marolt M. SymCHM: a compositional hierarchical model for pattern discovery in symbolic music representations. In: 11th Annual Music Information Retrieval eXchange (MIREX'15). Malaga; 2015. p. 1–3.
47. Lerdahl F, Jackendoff R. A generative theory of tonal music. Cambridge: MIT Press; 1983.
48. Sapp CS. Visual hierarchical key analysis. Computers and Intertainment. 2005; 3(4):1–19.
49. Woolhouse M, Cross I, Horton T. The perception of non-adjecent harmonic relations. In: Proceedings of International Conference on Music Perception and Cognition. Bologna; 2006.
50. Farbood M. Working memory and the perception of hierarchical tonal structures. In: Proceedings of International Conference of Music Perception and Cognition. Seattle; 2010.
51. Balaguer-Ballester E, Clark NR, Coath M, Krumbholz K, Denham SL. Understanding Pitch Perception as a Hierarchical Process with Top-Down Modulation. PLoS Computational Biology. 2009; 4(3):1–15.
52. Clarkson MG, Martin RL, Miciek SG. Infants' Perception of Pitch: Number of Harmonics. Infant behavior and development. 1996; 19(2):191–197. doi: 10.1016/S0163-6383(96)90018-1
53. Felleman DJ, Van Essen DC. Distributed Hierarchical Processing in the Primate Cerebral Cortex. Cerebral Cortex. 1991; 1(1):1–47. doi: 10.1093/cercor/1.1.1 PMID: 1822724
54. McDermott JH, Oxenham AJ. Music perception, pitch and the auditory system. Current opinion in Neurobiology. 2008; (18):452–463. PMID: 18824100
55. Leonardis A, Fidler S. Towards scalable representations of object categories: Learning a hierarchy of parts. Computer Vision and Pattern Recognition, IEEE. 2007; p. 1–8.
56. Fidler S, Boben M, Leonardis A. Learning a Hierarchical Compositional Shape Vocabulary for Multi-class Object Representation. arxivorg. 2014.
57. Engell A, Junghöfer M, Stein A, Lau P, Wunderlich R, Wollbrink A, et al. Modulatory Effects of Attention on Lateral Inhibition in the Human Auditory Cortex. PLOS ONE. 2016; 11(2). doi: 10.1371/journal.pone.0149933 PMID: 26901149
58. Di Russo F, Spinelli D, Morrone MC. Automatic gain control contrast mechanisms are modulated by attention in humans: evidence from visual evoked potentials. Vision Research. 2001; 41(19):2435–2447. doi: 10.1016/S0042-6989(01)00134-1 PMID: 11483175
59. Au WWL, Benoit-Bird KJ. Automatic gain control in the echolocation system of dolphins. Nature. 2003; 423(6942):861–3. doi: 10.1038/nature01727 PMID: 12815429
60. Emiya V, Badeau R, David B. Multipitch Estimation of Piano Sounds Using a New Probabilistic Spectral Smoothness Principle. IEEE Transactions on Audio, Speech, and Language Processing. 2010; 18 (6):1643–1654. doi: 10.1109/TASL.2009.2038819
61. Su L, Yang YH. Escaping from the Abyss of Manual Annotation: New Methodology of Building Polyphonic Datasets for Automatic Music Transcription. In: International Symposium on Computer Music Multidisciplinary Research; 2015.
62. Benetos E, Weyde T. An efficient temporally-constrained probabilistic model for multiple-instrument music transcription. In: Mueller M, Wiering F, editors. 16th International Society for Music Information Retrieval Conference. Malaga, Spain: ISMIR; 2015. p. 701–707.
63. Mirex. 2016: Multiple Fundamental Frequency Estimation & Tracking; 2016. Available from: http://www.music-ir.org/mirex/wiki.
64. Zeiler MD, Fergus R. Visualizing and Understanding Convolutional Networks. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T, editors. Computer Vision—ECCV 2014 SE—53. vol. 8689 of Lecture Notes in Computer Science. Springer International Publishing; 2014. p. 818–833.